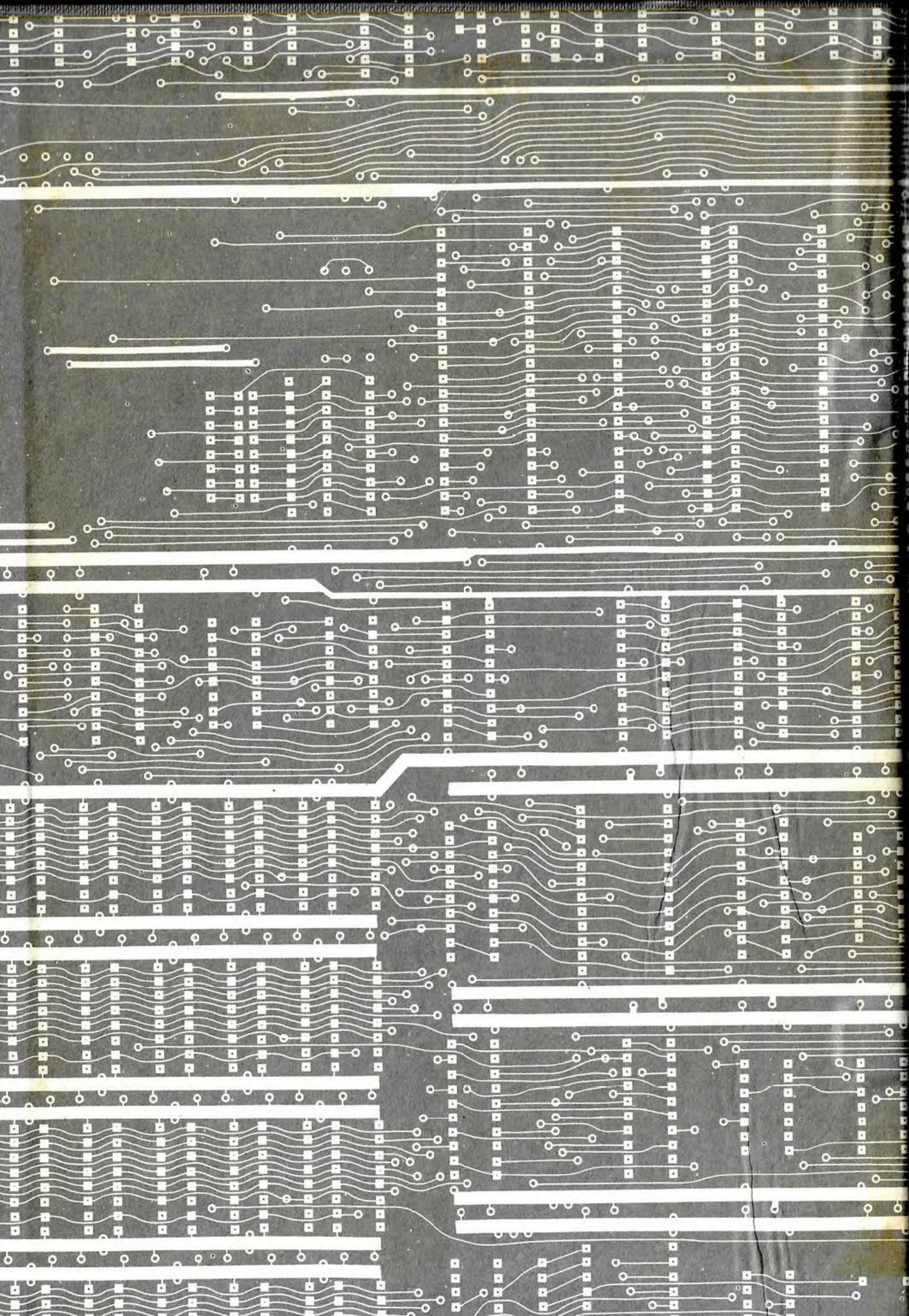
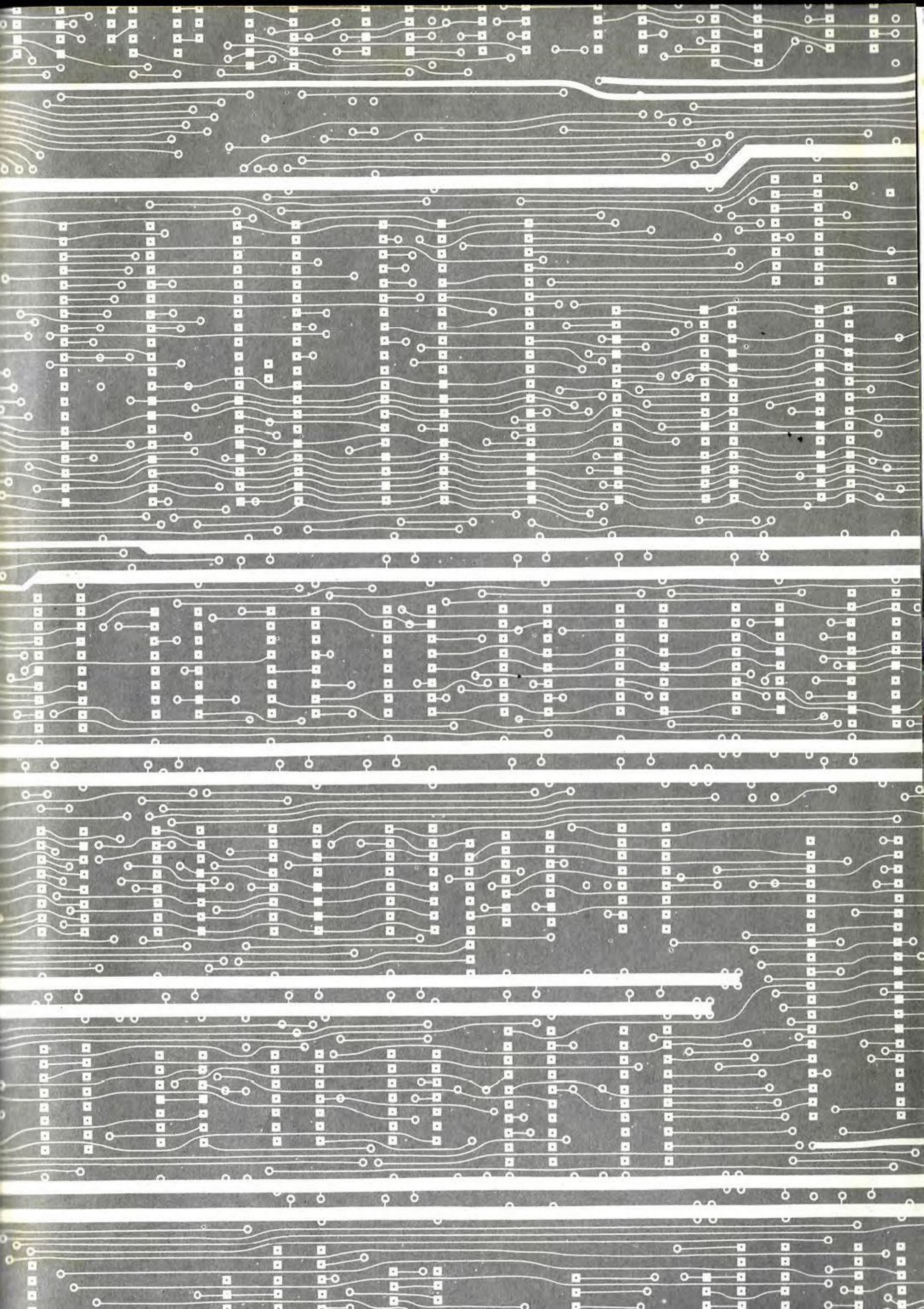
BASIC

ENCICLOPEDIA DE LA INFORMATICA
MINIORDENADORES Y ORDENADORES PERSONALES



EDICIONES FORUM



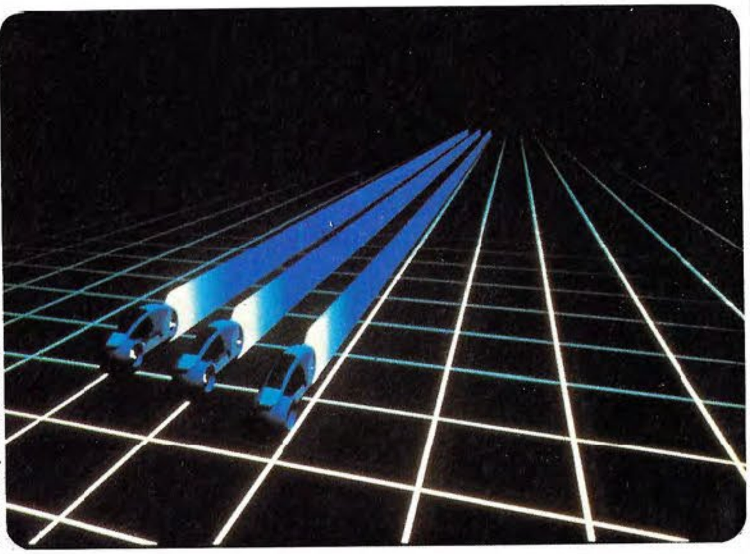


BASIC

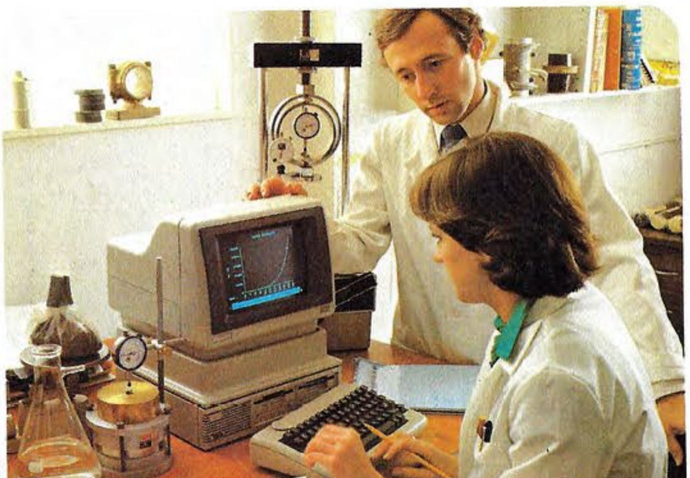
ENCICLOPEDIA DE LA INFORMATICA.
MINIORDENADORES Y ORDENADORES PERSONALES

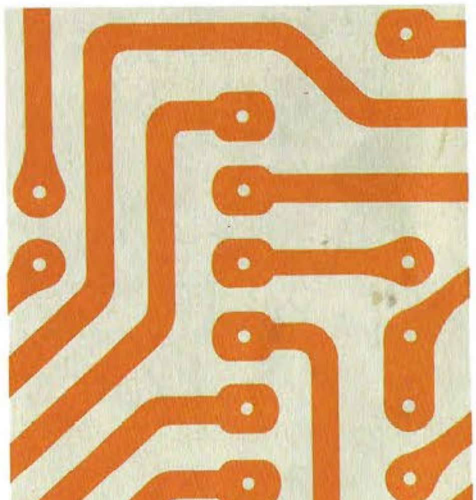
ENCICLOPEDIA DE LA INFORMATICA. MINIORDENADORES Y ORDENADORES PERSONALES

Walt Disney Productions ©



Hewlett-Packard





BASIC 1



BASIC

ENCICLOPEDIA DE LA INFORMATICA. MINIORDENADORES Y ORDENADORES PERSONALES

Presidente
José Manuel Lara

Director Ejecutivo
Jesús Domingo

Dirección editorial
R.B.A., Proyectos Editoriales, S.A.

Dirección técnica
Sante Senni

Con el asesoramiento de la Sociedad **E.G.S.**

REDACCION

Dirección: Gabriella Costarelli
Redactor jefe: Marcella Marcaccini
Secretaria de redacción: Giulia Abriani, Giovanna Aloisi
Revisión: Ugo Spezia
Corrección: Maria Albergo, Laura Salvini, Graziella Tassi
Recopilación material gráfico: Carla Bertini, Rossella Pozza
Producción: Piergiorgio Palma
Secretaria de producción: Maria Rita Ciucci
Diseño y dirección artística: Vittorio Antinori
Jefe estudio gráfico: Roberto Sed
Compaginación: Alberto Berni, Riccardo Catani, Patrizia Fazio
Dibujos: Renato Lazzarini, Gianni Mazzoleni, Rolando Mazzoni
Traducción: Carlo Frabetti
Diseño cubierta: Neslé Soulé
Jefe de Producción: Ricardo Prats

Composición electrónica: ITC-Fototipo. Barcelona-Madrid.
Imprime: CAYFOSA - Carretera de Caldas, Km. 3,7
Santa Perpetua de Mogoda (Barcelona)

Depósito Legal: B. 37.099/83
ISBN (Obra completa): 84-7574-040-5
ISBN (Volumen I): 84-7574-045-6
ISBN (Fascículos): 84-7574-044-8
Impreso en España - Printed in Spain

El editor agradece la colaboración de:
Alfa Romeo, Buffetti Data, Commodore Italiana, CPT Italia, Creazioni Walt Disney, Data General, Digital, Doxa, Elsag, Ericsson, Facit Data Products, Ferrari, FIAT, Harden Italia, Hengstler Italia, Hewlett Packard, IBM, Intema, IRET Informática, Italcable, Lancia, Litton BEI, MEE, MSI Data Italia, Olivetti, Perkin-Elmer, Plessey Trading, Prime Italia, Rank Xerox, Rhône-Poulenc Italia, Sanco Ibex Italia, Sarin, Selca Elettronica, Selenia, Sperry, SIP, Telespazio.

PRESENTACIÓN

Hasta hace poco la informática era una ciencia para iniciados, y los primeros ordenadores eran máquinas que servían para realizar cálculos a gran velocidad y que, con su aspecto imponente, rodeadas de hombres con bata blanca, estaban reservadas solamente a los más importantes centros industriales y de investigación. Hoy día las computadoras forman parte de nuestra vida y su cultura es, cada vez más, nuestra cultura; los ordenadores están en las universidades, en las fábricas y oficinas, desempeñan un importantísimo papel en las telecomunicaciones, conducen trenes y pilotan aviones, realizan las funciones de bibliotecarios, archivadores, traductores, se los encuentra en los quirófanos, prevén el tiempo e incluso juegan al ajedrez.

Nos hallamos, pues, en el umbral de una nueva era en la cual las computadoras, hasta ahora privilegio de unos pocos especialistas, estarán a disposición de todo el mundo.

Según una reciente investigación de mercado, en Estados Unidos las ventas de ordenadores personales aumentan en un cincuenta por ciento cada año, y cabe prever que, hacia el año 2000, habrá al menos trescientos millones de miniordenadores.

Probablemente, dentro de unos años un sistema "home computer", es decir, una computadora casera provista de monitor e impresora, no valdrá mucho más que un aparato de vídeo normal y estará, por tanto, enormemente difundido; mediante el ordenador doméstico podremos no sólo trabajar, estudiar, jugar, consultar datos y programar la vida cotidiana, sino que el sistema mismo de la información y de la enseñanza experimentará una auténtica revolución debido a la difusión de las computadoras.

Usar un ordenador significa, sin embargo, hablar su lenguaje, estar en condiciones de hacerle preguntas, de plantear las cuestiones de la manera más correcta y comprender las respuestas.

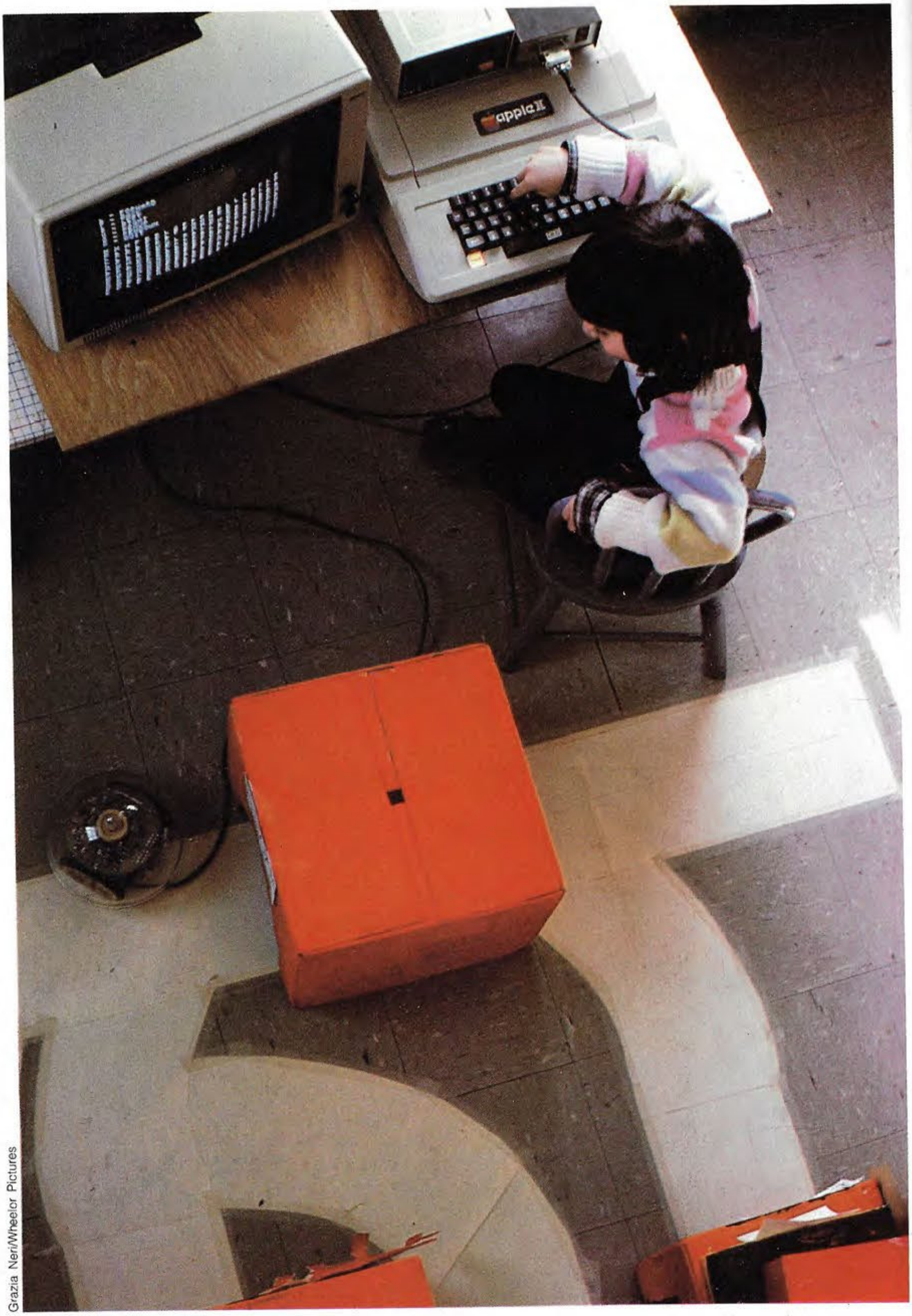
Han sido precisamente estas consideraciones las que nos han impulsado a publicar una obra como BASIC, Enciclopedia de la informática, de los miniordenadores y ordenadores personales, con la que pretendemos suministrar al lector no sólo la orientación necesaria para la elección del ordenador más adecuado a sus exigencias, sino sobre todo un instrumento indispensable para aprender el lenguaje, la simbología y las aplicaciones de las computadoras.

Deliberadamente elaborado como un curso didáctico, con abundantes dibujos e ilustraciones, BASIC toma de la mano al lector y lo acompaña desde la descripción elemental de un ordenador hasta el planteamiento y realización de un programa complejo y personalizado.

Un curso, por tanto, adecuado para todos, tanto para los que se acercan por primera vez a la informática como para quienes, ya expertos, desean mejorar sus conocimientos para lograr una utilización óptima de su ordenador.

Es probable que dentro de algunos años quien no conozca el lenguaje necesario para "dialogar" con una computadora corra el riesgo de verse marginado, como hoy una persona que no sepa leer, escribir o utilizar el teléfono. Una obra como BASIC es, pues, el instrumento indispensable para conocer y comprender esta nueva ciencia, la informática, que seguramente influirá en el futuro desarrollo social tanto como, en el pasado, la invención de la imprenta.

EL EDITOR



Grazia Neri/Wheeler Pictures

SUMARIO

Cómo está hecho un ordenador

Para evaluar las distintas características y las posibilidades de los diversos sistemas ofrecidos por un mercado en continua expansión, es necesario conocer la lógica de un ordenador, cuáles son las funciones que puede realizar y cómo tienen lugar en su interior.

El ordenador, un célebre desconocido
Qué es un programa y por qué resulta útil una máquina programable. Los datos: definición y uso. Pasado y futuro del cálculo automático: del ábaco a los primeros intentos de realizar una máquina calculadora. Las calculadoras. La idea de «programación»: nace el primer ordenador. Desarrollo de los ordenadores: ¿llevará la miniaturización a las máquinas «pensantes»?

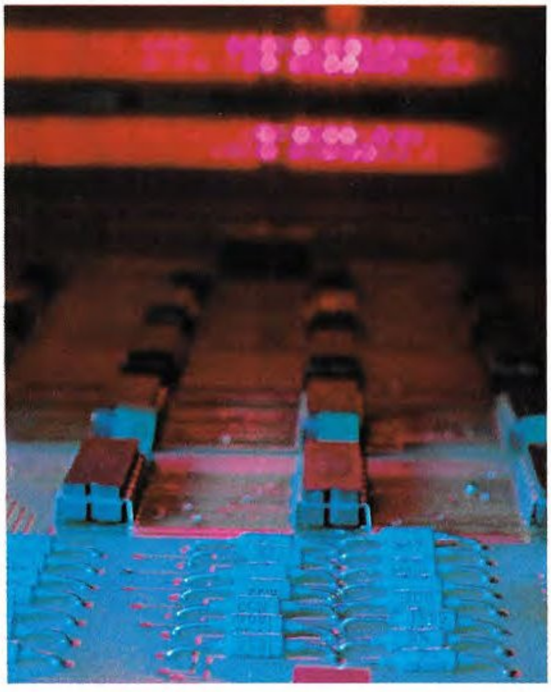
La lógica del ordenador
Cómo «razona» un ordenador. Máquinas analógicas y digitales. Báscula electrónica: un primer ejemplo de aplicación. Cómo se clasifican los ordenadores. Redes de transmisión de datos, conexiones entre ordenadores. Estructura de un centro de cálculo.

Los sistemas de numeración
Los impulsos: definiciones y ejemplos. Sistemas de numeración en el mundo de los ordenadores. Traducción de los números a las diversas bases. Operaciones aritméticas en base 2. Tablas de conversión de los números en los sistemas binario, octal y hexadecimal.

Elementos de lógica
Los operadores lógicos: qué son y cómo se utilizan. Los conjuntos. Aplicaciones de los operadores lógicos: búsqueda y selección en los archivos. Circuitos integrados y formas de onda. Aplicaciones de algunos circuitos integrados. Ejemplos de circuitos integrados que realizan las funciones lógicas. La seguridad de los datos: creación de los códigos con los operadores lógicos.

Códigos de transmisión de datos
Los códigos: traducción de nuestros símbolos gráficos al lenguaje máquina. Descripción del código ASCII. Los códigos transparentes, la codificación de letras y números. Generación de códigos de seguridad. Sistemas de transmisión de datos. Procesos sincrónicos y asincrónicos.

Sistemas de microordenador
Estructura de un sistema de microordenador. Unidades principales. Los BUS de conexión. Periféricos, métodos y procesos de transmisión de datos. Función de la unidad central (CPU) y cálculos que puede realizar. Funcionamiento de un sistema con microprocesador: ciclos e instrucciones. Contadores de programa, registros, stack. Utilización de la memoria; ejemplos de los ciclos de una instrucción. Los saltos en el seno de los programas. Interacción con el mundo exterior: el Interrupt. Estado de Hold. Transferencia de datos con la técnica DMA.



Archivio Curcio/S. Petrelli

Traducción de un problema práctico en un esquema lógico

Qué es un programa. Métodos y simbología para racionalizar y resolver un problema práctico. Presentación básica de los principales lenguajes simbólicos. Los métodos de memorización de los datos y la lógica a seguir para la documentación de los archivos.

La programación
Los lenguajes de programación. Lenguajes compilados e interpretados. La compilación. Principales tipos de lenguaje de alto nivel. Los lenguajes de bajo nivel.

Análisis y diagramas de los flujos
Planteamiento de los programas: análisis del problema y de los datos disponibles. Esquematización del problema y métodos de subdivisión. Aplicaciones: definición de los procedimientos de gestión de un almacén. Diagramas de flujo: definiciones y simbología. Uso de los símbolos. Los índices y los flags. Loops y contadores: métodos de empleo, ejemplos de aplicaciones. Las dos clases principales de aplicaciones: gestionarias y científicas. Principales diferencias en la estructuración de los programas. Formulario geométrico. Formulario financiero.

Archivo de datos
Principales tipos de memoria de masa. Los archivos. Estructuración de los archivos para la utilización de micro y miniordenadores. Los files y los records. Los distintos tipos de files: principales características y campos de aplicación. Estructuración y organización de datos. La documentación de los archivos. Archivos complejos. Estructuras fundamentales de los archivos. Los bancos de datos.

Planteamiento de los programas
Presentación de los principales ordenadores personales y miniordenadores. Tablas comparativas y criterios de valoración. Valoración de los sistemas operativos. Segmentación de los programas en función de la complejidad del problema. Criterios de elección del aparato.

El lenguaje de programación Basic

Teorías y métodos para programar en Basic: el funcionamiento del «modo inmediato»; qué son las funciones matemáticas y cómo se desarrollan en Basic; metodologías para el correcto empleo de las instrucciones Basic. Métodos de división de los programas.

Generalidades
Utilización del Basic interpretado para el desarrollo de los cálculos: el Basic en el «modo inmediato». Constantes y variables, definiciones, precisión en el cálculo. Representación de los números reales en forma exponencial. Representación de los números enteros y de los reales en el interior del aparato. El área de memoria: memoria de programa y memoria de datos. Los cálculos: prioridades, precisiones y redondeo en el desarrollo de los cálculos. Las cadenas, qué son y cómo se utilizan. Operaciones con cadenas. Diferencias entre las precisiones obtenibles con los distintos tipos de ordenador personal. Las subcadenas.

Los mandos
Los mandos del Basic. Escritura y conservación de los programas. Edición de datos.



Honeywell



Las instrucciones TRON y TROFF. Utilización de estas instrucciones en la búsqueda de errores.

Las instrucciones
Instrucciones de asignación para las variables numéricas y las de cadena. Los loops: su utilización y sintaxis; principales errores de los loops. Los saltos: saltos de instrucción, llamada a las subrutinas. Utilización de los operadores lógicos en la programación. Las instrucciones condicionales. Aplicaciones: planteamiento de un programa para el cálculo de las amortizaciones.

Las funciones
Concepto de función. Representación gráfica. Funciones numéricas en Basic: su utilización, límites y precisión. Funciones numéricas definidas por el usuario. Funciones de cadena en Basic. Notas sobre estadística: generación de números aleatorios. Aplicaciones: planteamiento y escritura de un programa para la resolución de ecuaciones; planteamiento y escritura de programas para la gestión de las cadenas de caracteres. Formulario matemático: funciones.

Los array
Memorización de datos. Definición de los array mono y multidimensionales. Array numéricos y de cadena. Las instrucciones de asignación en los array. Aplicación: planteamiento y escritura de un programa para efectuar cálculos con un número cualquiera de cifras; uso de los array para la generación de los mensajes diagnósticos de error.

Funciones de entrada y salida de datos
Instrucciones de lectura e impresión de datos. El cursor: mandos del vídeo. Uso de la impresora, códigos para las funciones concretas. Las teclas programables: las instrucciones de programación y su uso. Los MENU, sistemas de presentación de los programas de un proceso. Las instrucciones de manipulación de los bits. Instrucciones para el acceso directo a la memoria.

Gestión de archivos
Formatos de memorización de datos en disco. Codificaciones y transformación de los datos. Las instrucciones de lectura y escritura hacia el disco. Funciones numéricas y de cadena en correlación con el disco. Métodos de gestión de los archivos múltiples. La gestión de archivos complejos: descripción de algunos programas para la gestión de datos. Ordenación de archivos; problemas ligados a esta función. Programas de utilidad para la ordenación de archivos. Búsqueda y selección de datos: las subrutinas de comparación. Aplicaciones: planteamiento y escritura de un programa de gestión de una agenda; planteamiento y escritura del programa para la gestión de un almacén; planteamiento y escritura de la subrutina para redondear los cálculos.

Estructuración de los programas
Principios de estructuración de los programas. Técnica de aproximación Top-Down. Generalización de las subrutinas. Segmentación de los programas. Transferencia de datos entre programas o entre segmentos; uso del disco como memoria auxiliar. Máscaras de vídeo, ejemplos de aplicaciones prácticas.

Conclusiones
Métodos de validación de los programas. Búsqueda de errores. Compendio de las instrucciones Basic. Aplicaciones: uso de varios files en un mismo programa. Planteamiento y preparación de los diagramas de flujo de primer nivel para un proceso de facturación.

La programación avanzada

Otros tres importantes lenguajes de ordenador: Assembler, Cobol, Fortran. Examen en profundidad de las instrucciones en Fortran, comparadas a las del Basic, y sus aplicaciones científicas. Técnicas de programación con lenguajes mixtos. Empleo del ordenador para la realización de gráficos y dibujos. Posibles aplicaciones, especialmente en el campo del control automático de maquinaria e instrumental complejos.

Assembler
El lenguaje Assembler, estructura y principales características. Formato de los principales tipos de instrucción en Assembler. Algunos ejemplos de aplicación. El Macroassembler. Empleo del Assembler; cuándo y por qué puede resultar útil. Aplicaciones: proceso de facturación, preparación de los diagramas de flujo de detalle.

Cobol
El lenguaje Cobol. Estructura y características. Algunos ejemplos de instrucciones.

Fortran
Estructura del lenguaje Fortran. Campo de aplicación. Las instrucciones en Fortran. Comparación entre las instrucciones en Fortran y las instrucciones en Basic. Aplicaciones: facturación; escritura de programas.

Programas mixtos
Sistemas de programación en varios lenguajes. Uso de programas mixtos. Basic compilado: diferencias y ventajas con respecto al Basic interpretado. Sistemas operativos, estructura y funciones; programas de utilidad en los sistemas operativos. Sistemas operativos en multiprogramación.

Gráficos y ordenadores
Gráficos, histogramas, escalas de representación. Subrutinas para la preparación de histogramas y gráficos. El videográfico. El plotter. Utilización de la impresora para la preparación de histogramas. Aplicaciones: escritura de subrutinas; escritura de subrutinas para la preparación de histogramas. Representación gráfica de funciones.

Aplicaciones especiales
Una ojeada al futuro: desarrollo de los sistemas con microprocesador. Esquema de utilización de los microordenadores en los controles automáticos.

Glosario general

Índice analítico



GLOSARIO BÁSICO

ASCII. El principal sistema de codificación para el intercambio de datos entre ordenadores.

Bit. Mínima cantidad de información; puede valer 1 o 0.

Bus. Conjunto de circuitos eléctricos por los que viajan las señales (datos, instrucciones o controles).

Byte. Agrupación de bits (normalmente 8) que forma una instrucción.

Campo. Cada una de las zonas «lógicas» en que se divide un record. Los campos pueden ser numéricos, si contienen sólo números, o alfanuméricos, si contienen también letras.

Clave. Valor particular de uno de los campos en que está dividido un record, que permite el acceso a los datos. En una agenda telefónica, por ejemplo, es el apellido.

Código transparente. Se llama así a los códigos ASCII a los que no corresponden signos. Se usan para enviar órdenes, o respuestas a las órdenes.

Compilador. Programa que forma parte del sistema operativo, y que traduce los programas fuente al lenguaje comprensible para la máquina.

CPU. Unidad central de proceso. Es el circuito que gobierna el funcionamiento de todo el sistema del microordenador.

Cursor. Punto de la pantalla de vídeo, normalmente visualizado con un cuadradito luminoso, a partir del cual el aparato empieza a emitir los mensajes o a aceptar los datos de entrada. En términos más generales, es el punto de la pantalla en el que se insertará el signo siguiente.

Dirección. Es un número que identifica una memoria. En otras palabras, para leer o escribir en una memoria es necesario especificar a qué memoria se refiere la instrucción, y el número de referencia se llama dirección. En aparatos pequeños y medianos las direcciones, y por tanto las memorias, tienen valores comprendidos entre 0 y 64.000 aproximadamente; en los más grandes pueden superar el millón.

Display. Dispositivo de visualización (por ejemplo, una pantalla de vídeo).

DMA. Método de transferencia de datos con acceso directo a la memoria.

File. Cada una de las zonas de la memoria de masa (floppy) en las que se memorizan informaciones homogéneas (archivos, etc.).

Flag. Variable cuyo valor depende de que se den o no determinadas situaciones. Sirven para memorizar si la situación se dio o no.

Floppy. Disco inamovible. Su capacidad, en caracteres, varía entre 160.000 y 1.200.000.

Flujo (diagrama de). Representación gráfica de las operaciones que se han de realizar en un programa. El «flujo» es la indicación del camino que sigue el aparato para realizar la tarea que se le ha encomendado.

Fuente (programa). Cualquier programa escrito en un lenguaje simbólico.

Hold. Situación de espera de la CPU para completar una operación.

Impulso. Señal eléctrica de corta duración. Un impulso puede corresponder a un bit.

Intercalación. Operación de unión de programas o datos gracias a la cual se puede unir varios files o varias partes de programa para obtener uno solo.

Interrupt. Interrupción del programa por una causa externa.

I/O (Input/Output). Función genérica de entrada (Input) o salida (Output).

Linker. El programa que, por compilación, «reúne» las diversas partes del programa general para obtener el programa objeto, listo para su ejecución.

Loop (Lazo). Parte repetitiva de un programa, que puede efectuarse un determinado número de veces fijado de antemano.

Objeto. Un programa compilado y listo para ser utilizado.

Record. Cada una de las zonas «lógicas» en que puede dividirse un file. En un file que contenga direcciones, por ejemplo, el record puede ser cada una de las direcciones mismas.

Salto (JUMP). Una instrucción que puede aislar, «saltándosela», una parte del programa.

Segmentación. Técnica mediante la cual se dividen los programas muy amplios en partes más pequeñas y por tanto más fáciles de utilizar.

Sort. Programa especial que sirve para ordenar (por ejemplo, alfabéticamente) los datos.

Stack. Área de memoria reservada a las «direcciones de retorno». Durante los saltos sirve para conservar la dirección, es decir, la instrucción, de la que se ha partido. De esta manera, tras el salto, se conserva la memoria del punto de partida, y se puede (si es necesario) volver atrás.

Subrutina. Parte de un programa que realiza una función específica. Las subrutinas pueden ser utilizadas desde cualquier punto del programa principal.

Top-Down. Técnica de programación que consiste en el desarrollo del análisis y de la codificación de las instrucciones, partiendo de una visión global del problema y subdividiéndolo en un determinado número de problemas más simples.

El ordenador, un célebre desconocido

La principal función del ordenador es la de realizar todas las actividades que no requieran una participación inteligente, pero que posean, en cualquier caso, un grado de complejidad decididamente superior al simple desarrollo de cálculos. Son, por tanto, aparatos capaces de estructurar y elaborar información. Por esta razón el nombre de «ordenador» ha acabado imponiéndose sobre el de «calculador electrónico» o «computadora», aunque en la práctica todos estos términos se usan como sinónimos.
La diferencia esencial entre un ordenador y una máquina de calcular estriba en la capacidad de «aprendizaje» que tiene el primero. A estos aparatos cabe «enseñarles» cómo llevar a cabo determinadas tareas, es decir, pueden ser «programados». El de los ordenadores es, sin embargo, un «aprendizaje» pasivo: no pueden contribuir a él en modo alguno.
Este tipo de instrucción, es decir, la «programación», es muy similar a los reflejos condicionados.
Por ejemplo, podemos adiestrar a nuestro perro para que vaya al quiosco, se haga entregar el periódico y nos lo traiga; estas acciones están guiadas por reflejos condicionados generados por el adiestramiento, sin ninguna participación inteligente; si el perro encuentra el quiosco cerrado o si nuestro periódico no ha salido, no puede elegir una acción alternativa para la que no ha sido adiestrado.
El mismo concepto es válido para el «adiestramiento» (la programación) de los ordenadores; pueden realizar todas las funciones para las que hayan sido programados, pero ante cualquier acontecimiento no previsto es necesaria la intervención humana. En el futuro, si las investi-

El ordenador, en la gestión de una biblioteca, permite efectuar búsquedas por tema y por autor en tiempos brevísimos.



Grazia Neri/Wheeler Pictures

gaciones de vanguardia, ya iniciadas, dan resultados positivos, podrían construirse aparatos con una cierta forma de «autoconciencia»; pero hoy día esta posibilidad pertenece todavía al campo de la ciencia ficción, aunque ya existen ordenadores a prueba de error, es decir, en condiciones de detectar eventuales averías, aislar los circuitos defectuosos y derivar sus funciones hacia los circuitos de reserva. Sus capacidades actuales y su posible desarrollo no deben, sin embargo, inducirnos a pensar en los ordenadores como aparatos extremadamente complejos y reservados a los especialistas. Los ordenadores pueden ser usados por todo el mundo, especialmente los de tipo más reducido (ordenadores personales), y representan la evolución natural de las máquinas de calcular.

Dispositivo especial, denominado "pluma electrónica", que permite leer directamente las informaciones del vídeo.



Marka

Efectivamente, el uso de la calculadora se ha vuelto ya indispensable en cualquier actividad; el profesional, el comerciante o el estudiante encuentran natural hacer efectuar los cálculos a un aparato y reservar su tiempo a las actividades que exigen capacidad de pensamiento.
El estudiante, para resolver una tarea de matemáticas, primero tiene que plantear el problema y luego efectuar los cálculos; esta segunda parte no es significativa con respecto a la racionalización y subsiguiente solución y, por tanto, puede ser perfectamente realizada por un aparato.
Si, por ejemplo, un estudiante quiere hallar la hipotenusa de un triángulo rectángulo conociendo los catetos, ha de indicar la solución recurriendo al teorema de Pitágoras:

a) hipotenusa $= \sqrt{(\text{primer cateto})^2+(\text{segundo cateto})^2}$

Si los catetos valen:
primer cateto=8
segundo cateto=4
la fórmula para la solución, obtenida de la anterior sustituyendo las letras por los valores numéricos, es:

b) hipotenusa $= \sqrt{8^2+4^2}=\sqrt{64+16}=\sqrt{80}\simeq 8{,}9$

Para encontrar el valor y concluir el ejercicio, el estudiante tendrá, pues, que extraer la raíz cuadrada de un número; la generación anterior buscaba el resultado en las tablas, la actual utiliza la calculadora y la próxima usará el ordenador. Veamos por qué.
La solución del problema está en la fórmula (a), y el paso sucesivo (b) es necesario para hallar el valor numérico. El estudiante que haya resuelto la primera parte (a) demuestra que conoce el problema y sabe plantearlo correctamente; la fase siguiente de cálculo (b) no es más que un complemento engorroso y no significativo, y su realización se puede delegar a una máquina.
Si se dispusiera de una máquina a la que se pudiera decir:
toma el valor del primer cateto y elévalo al cuadrado (8^2);
toma el valor del segundo cateto y elévalo al cuadrado (4^2);
suma los dos cuadrados precedentes (64+16);
saca la raíz cuadrada de la suma ($\sqrt{80}$);
da el resultado (8,9),



Marka

En las escuelas, el ordenador "docente" sigue a los alumnos con cursos personalizados.

tras la fase de planteamiento el estudiante no sólo habría terminado su trabajo, sino que para cualquier otro problema en el que tuviera que aplicar el teorema de Pitágoras tendría ya lista la solución.

Los datos

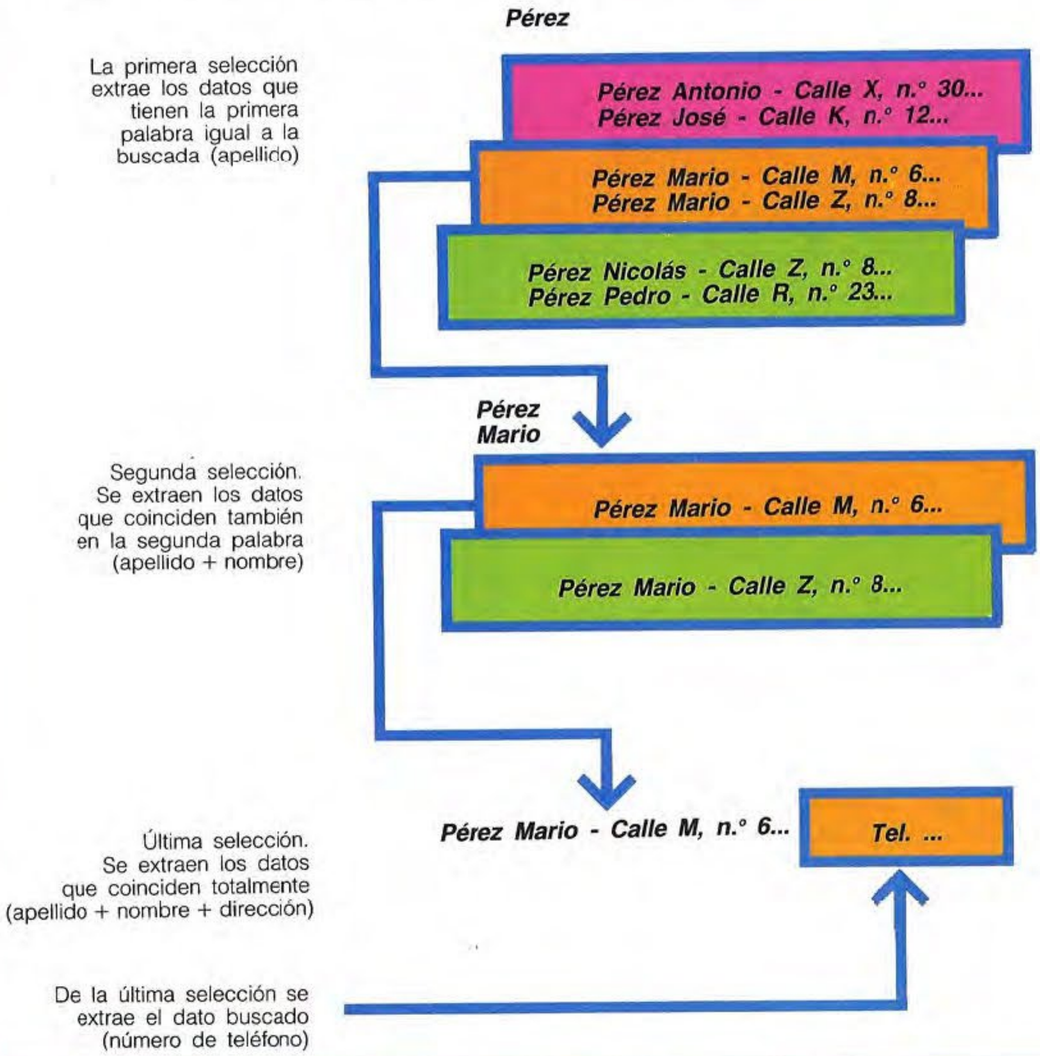
El aparato que puede realizar estas funciones es el ordenador, al que podemos «enseñar», programándolo, la forma de resolver el teorema de Pitágoras. Nuestra tarea consiste en suministrarle, la primera vez, todas las indicaciones necesarias para la realización de ese cálculo; las veces siguientes sólo tendremos que dar los nuevos valores de los catetos y el ordenador, conociendo ya las operaciones a efectuar, podrá suministrar rápidamente el resultado.
Cuando realizamos un cálculo, por ejemplo una suma, conocemos los dos sumandos y queremos obtener el resultado. Los dos números a sumar son los datos del cálculo.
Si buscamos un número de teléfono en la guía, necesitamos saber el nombre y apellido y la dirección del abonado. En la búsqueda utilizamos esta información como datos para obtener el número. En este caso la acción (búsqueda del número) termina con la lectura del número mismo; también éste es un dato, en cuanto que ya está contenido en la guía.
En el primer caso *(cálculo)* se obtiene un valor *(resultado)* que, a priori, no existía; en el segundo *(búsqueda)* se obtiene un valor preexistente.
Estas dos formas principales de utilización de datos están esquematizadas en el gráfico de la página 16.
Con los ordenadores hay que seguir la misma lógica. Para realizar un cálculo se han de conocer todos los valores numéricos (datos numéricos) que intervienen en el cálculo; para una búsqueda hay que conocer todos los datos en base a los cuales se efectúa la selección.
Cuando hablamos con alguien, intercambiamos informaciones mediante una serie de datos. Tales datos pueden ser tomados de nuestra memoria, como por ejemplo cuando referimos un acontecimiento, o bien pueden ser el resultado de una elaboración, si expresamos una opinión.

UTILIZACIÓN DE LOS DATOS DEL ORDENADOR

Los datos se utilizan para suministrar un resultado



Los datos se utilizan para efectuar elecciones



Para referir un acontecimiento o expresar una opinión hay que conocer las circunstancias. **Análogamente el ordenador, para suministrar cualquier respuesta, ha de poseer todos los datos.**

Por ejemplo, observando el agua que hierve podemos deducir que, si la tocamos, nos quemaremos. Este proceso mental es posible en la medida en que, previamente, alguien nos ha advertido de ese peligro, hemos almacenado la información y, al darse la situación que se nos ha descrito, estamos en condiciones de recordar la advertencia. El razonamiento que hacemos es, pues, un proceso asociativo entre dos datos: agua hirviendo y peligro de quemarse.

Este mismo procedimiento se aplica en los ordenadores: inicialmente es necesario suministrar todos los datos y las eventuales reglas de asociación; posteriormente, ante una situación dada (por ejemplo, una demanda por parte del operador), el aparato puede determinar la respuesta correcta.

Como sabemos, sin embargo, nuestras posibilidades de almacenamiento de datos en el cerebro no se limitan a la transferencia directa por parte de alguien que nos instruye; aprendemos también a través de nuestras experiencias. Volviendo al ejemplo del agua hirviendo, en el caso de que nunca nos hubieran advertido, la primera vez hubiésemos tocado el recipiente y nuestros receptores (terminaciones nerviosas) habrían enviado sensaciones de dolor al cerebro, con lo que la situación hubiera sido catalogada como peligrosa. Tras esta primera vez, habríamos memorizado la experiencia necesaria para establecer el comportamiento futuro ante la misma situación.

El método de aprendizaje, que nos suministra continuamente nuevos datos y enriquece nuestras posibilidades, ha sido aplicado también a los ordenadores; es una aplicación todavía en sus comienzos, pero que ya produce excelentes resultados y permite entrever amplias perspectivas.

En los ejemplos ofrecidos están implicadas las dos formas posibles de razonamiento:

- **concepción determinista**
- **concepción probabilística**

La concepción determinista consiste en asociar causa y efecto mediante una gran cantidad de datos introducidos en el ordenador, de forma que estén previstas todas las situaciones.



Ericsson

Un ordenador personal con teclado ergonómico, diseñado racionalmente.

En la concepción probabilística la lógica es más compleja y puede explicarse mejor con un ejemplo.

Supongamos que queremos enseñar a un ordenador a jugar al ajedrez.

Utilizando el método determinista, tendremos que introducir en su memoria una respuesta para cada situación posible; un proceso de este tipo, además de extremadamente trabajoso, resultaría casi con seguridad incompleto.

La otra alternativa consiste en memorizar las reglas de acuerdo con las cuales pueden moverse las distintas piezas y hacer que el ordenador juegue un gran número de partidas. En cada partida, si una jugada (hecha inicialmente de forma casual) produce buenos resultados, el ordenador memoriza la situación y el movimiento efectuado es clasificado como válido; si, por el contrario, el resultado es negativo, la situación es clasificada como algo a evitar. Naturalmente, el ordenador perderá al principio todas las partidas, pero, a medida que aumente su experiencia, aumentará el número de situaciones reconocibles y será cada vez más difícil ganarle. Con este sistema, el ordenador puede «aprender».

¿Puede pensar un ordenador?

El científico estadounidense Robert Jastrow, haciendo una comparación entre las diferentes características del cerebro humano y el ordenador, en su libro *El telar encantado* –del que a continuación reproducimos un fragmento– ha formulado algunas teorías sobre la capacidad de «pensamiento» de las computadoras.

Nadie se atrevería a negar la superioridad de los ordenadores en lo que respecta a la aritmética y los cálculos matemáticos en general, pero difícilmente se admite que puedan hacer algo más. ¿Cómo puede una máquina pensar o razonar? La cosa parece imposible, pero un interesante programa de ordenador, elaborado por A.L. Samuel, de la IBM, ilustra la forma en que tal cosa podría suceder. El programa, que enseña a un ordenador a jugar a las damas, demuestra que la máquina es capaz de expresar razonamientos próximos a los humanos, si ha sido oportunamente predispuesta.

Samuel empezó el adiestramiento de su ordenador enseñándole las reglas del juego de damas, que la máquina almacenó en su memoria. Luego le confirió la prerrogativa humana de la experiencia de juego facilitándole una fórmula que permitía al ordenador evaluar la fuerza de sus posiciones en el tablero. ¿Cómo era posible tal cosa? En primer lugar, la fórmula analizaba la posición de las fichas por separado, y luego sumaba sus ventajas y desventajas, tales como el número de posibilidades de «comer» una ficha contraria o las eventuales trampas en que podían caer las propias. La fórmula incluía también estimaciones más sutiles; por ejemplo, aconsejaba las jugadas que provocaban el intercambio de fichas en el caso de que la máquina fuera ganando; si, por el contrario, iba perdiendo, la fórmula desaconsejaba el cambio de piezas. Cualquier jugador humano sigue la misma estrategia. Finalmente, la fórmula expresaba la fuerza de la posición de la máquina con un número: cuanto mayor era el número, más fuerte la posición. La fórmula era para la máquina el equivalente de la experiencia de un jugador humano. Samuel había transmitido, en la práctica, a la máquina el saber de la humanidad, del mismo modo que un maestro se lo transmite a un alumno. Normalmente, los jugadores humanos no tienen fórmulas en la mente, pero basan su estrategia en un conjunto de reglas empíricas tomadas de la experiencia, esencialmente equivalentes a una fórmula.

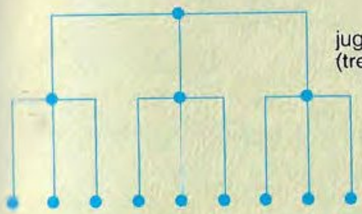
Tras enseñar a la máquina los fundamentos de la estrategia del juego de damas, Samuel la programó de forma que a medida que jugaba lo hiciera cada vez mejor; es decir, introdujo en ella la capacidad de aprender en base a la experiencia. Para el ordenador eso significaba actualizar los datos de la fórmula después de cada jugada. ¿Cómo podía la máquina superar en habilidad a su maestro? Comparaba el valor que esperaba de una jugada con el valor real obtenido tras la respuesta del adversario. La diferencia se debía, naturalmente, al hecho de que el adversario podía no hacer la jugada que la máquina presuponía. La comparación revelaba a qué factores de la fórmula se tendría que haber atribuido un peso mayor o menor. La máquina modificaba estos «pesos» en la fórmula después de cada jugada, de manera que las fórmulas cambiaban continuamente y la máquina mejoraba su estrategia de juego. Igual que los seres humanos, aprendía con la práctica.

¿Cómo se juega contra un ordenador? Al comienzo de la partida, se dispone el tablero con las piezas, que luego alguien moverá a cada jugada, o bien el tablero y las piezas aparecen en una pantalla de televisión. Todo ello en beneficio del jugador humano: la máquina no necesita mirar el tablero, puesto que tiene todos los datos en su memoria electrónica.

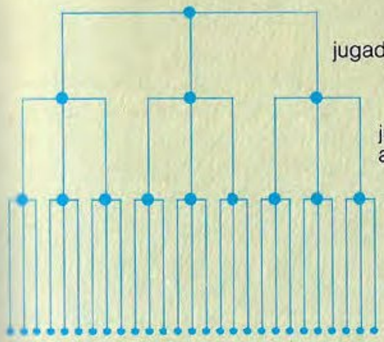
Los dos jugadores efectúan sus movimientos, uno tras otro. El jugador humano comunica sus jugadas a la máquina mediante un teclado. Cuando la máquina ha decidido su respuesta, hace aparecer el correspondiente signo en la pantalla, o bien lo imprime sobre papel. Remitiéndose continuamente a la fórmula, la máquina elabora cada jugada de la misma manera que un ser humano: intenta calcular los posibles desarrollos del juego tan lejos como puede, así como las ventajas relativas de cada jugada posible. Al hacerlo, parte de la hipótesis de que el adversario contestará con la mejor jugada desde su punto de vista.

El método del ordenador al jugar a las damas es muy similar al de un principiante que calcula mentalmente las propias posibilidades, paso a paso: «Si muevo esta pieza, él moverá tal otra. Si muevo aquella, él moverá...». La diferencia entre el principiante y el jugador experto estriba en que este último está familiarizado con muchas situaciones del juego y con las probables

reacciones del adversario, y puede prever el resultado de una jugada con mucha antelación, sin elaborar conscientemente las jugadas intermedias. Un observador exterior diría que el experto tiene un «sentido» intuitivo de las posiciones en el tablero. El experto elabora las jugadas de la misma manera que el ordenador.
¿Cómo «piensa» el ordenador? Supongamos que en un momento dado la máquina tenga tres jugadas posibles; entonces explora cada una de las posibilidades, calculando el valor de la jugada e intentando determinar la respuesta más probable del adversario. Supongamos que el contrincante también tiene tres respuestas posibles para cada jugada del ordenador. En tal caso, el ordenador tendrá que considerar un total de nueve posibilidades. La figura siguiente ilustra las alternativas que en este momento se ofrecen al ordenador. Cada bifurcación corresponde a una alternativa. El diagrama se conoce con el nombre de árbol de decisiones.



Anticipando las jugadas, la máquina contempla nueve posibilidades. Gracias a su memoria electrónica, las tiene en cuenta todas y elabora para cada una la jugada sucesiva. Esto significa añadir otra ramificación al árbol. Si sigue habiendo tres posibilidades para cada una de las nueve precedentes, tendremos en total 27 ramas.



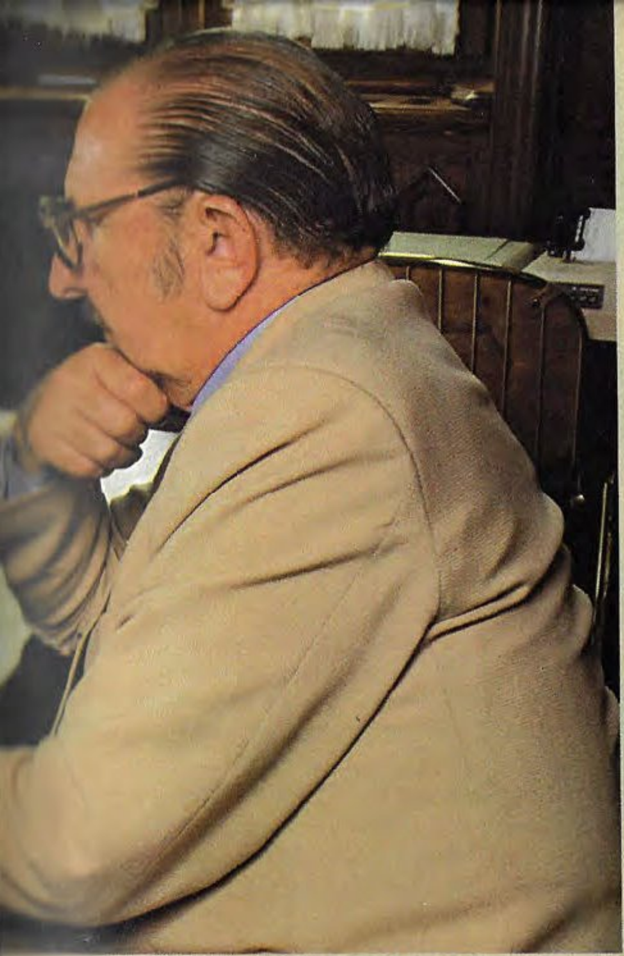
El siguiente estadio de este proceso convertiría en 81 el número de las posibilidades, lo que significa que la máquina debería tener en su memoria 81 tableros. La cosa empieza a ser dura, incluso para un ordenador, al menos para el viejo tipo de ordenador de válvulas del que disponía Samuel en los años cincuenta.
El programa de Samuel se detenía en este punto; sólo tras mucha insistencia la máquina proseguía. Hoy los ordenadores tienen una velocidad miles de veces superior, pueden anticipar más jugadas y juegan a las damas mucho mejor. De hecho, con los mismos principios se ha enseñado a los ordenadores el aún más difícil juego del ajedrez.
La sutileza de la estrategia del ajedrez confiere una cualidad realmente biológica a la inteligencia de un ordenador que sepa jugarlo. David Levy, campeón escocés de ajedrez –maestro internacional y uno de los 500 mejores jugadores del mundo–, tras jugar una partida con un ordenador considerado en su momento el mejor jugador artificial de ajedrez del mundo, ha dicho: «Uno está tentado de considerar que estos ordenadores son casi personas, sobre todo cuando ves que han comprendido lo que intentabas hacer y responden con combinaciones sumamente inteligentes».
Desde luego, el ajedrez es un juego muy abstracto. Una máquina con una lógica adecuada para el ajedrez puede, sin embargo, carecer de la inspiración creadora que caracteriza las más altas esferas del pensamiento humano. Igual que un joven recién graduado, tiene capacidades intelectuales prodigiosas pero le falta sabiduría. Sin embargo, también esta superioridad cualitativa del cerebro con respecto a los ordenadores está siendo cuestionada.
Estos nuevos desarrollos dependen en parte del aumento prodigioso del número de componentes que pueden comprimirse en un circuito integrado. Los circuitos integrados son como las neuronas del cerebro humano; conectados entre ellos en gran número, constituyen los modernos «cerebros electrónicos». Antes, hacia 1970, una plaqueta, o chip, podía incluir sólo un pequeño número de transistores y otros componentes electrónicos; por consiguiente, sólo podía «pensar» o recordar, pero no ambas cosas. Hoy día, los mejores fabricantes estadounidenses y japoneses de circuitos integrados han logrado condensar tantos componentes electrónicos en un circuito que, por vez primera, se puede combinar en una única plaqueta circuitos de elaboración y unidades de memoria. De lo que



Dominique Fradin

El ordenador juega al ajedrez; sus jugadas plantean un nivel de dificultad seleccionable.

se desprende que un circuito de este tipo es capaz tanto de recordar como de «razonar».
Esta combinación de funciones es enormemente importante en tanto que, por ejemplo, un circuito de memoria con capacidad de elaboración puede ser conectado de forma que envíe instrucciones a circuitos vecinos y reciba informaciones de ellos. Por tanto, cuando se envía una instrucción a las unidades de memoria del ordenador para que soliciten el contenido (por ejemplo, el nombre de una persona) almacenado en un circuito determinado, los circuitos de memoria directamente implicados pueden transmitir a los circuitos vecinos demandas de información y ofrecer al usuario una cantidad de información mayor que la que esperaba.
Nos acercamos así al tipo de memoria asociativa que tiene lugar en la corteza cerebral humana y que constituye un poderoso elemento del raciocinio. El cerebro responde a una simple demanda de información con una gran cantidad de material relacionado con la demanda concreta mediante asociaciones que se remontan a la experiencia pasada. El pensamiento humano viene facilitado enormemente por la riqueza de las respuestas del cerebro a tales demandas, respuestas que se desarrollan a nivel inconsciente a través de las miles de conexiones entre las neuronas. Un ordenador que tuviera los mismos circuitos, conectados cada uno a un gran número de otros circuitos, sería un verdadero «cerebro de silicio». Al igual que el cerebro humano, funcionaría como una sola unidad, operando mediante ondas de actividad mental interna a nivel «inconsciente».
Los últimos progresos de la tecnología de los circuitos integrados hacen posible también otro tipo de pensamiento creativo en los ordenadores. El pensamiento de los ordenadores ha tendido a ser mecánico y carente de imaginación porque las conexiones entre sus circuitos son mucho más simples que las del cerebro. Cada puerta lógica de un ordenador tiene sólo dos o tres cables procedentes de otras partes de la máquina, mientras que una microscópica célula o puerta lógica del cerebro humano tiene decenas de millares de «cables», o sea, de fibras nerviosas, procedentes de otras partes del ce-



rebro. Las innumerables conexiones entre una célula y otra, combinadas con las sofisticadas características de las puertas lógicas propias del cerebro, explican en gran medida las extraordinarias capacidades de este órgano.

En teoría, los ordenadores podrían haber sido construidos hace mucho tiempo con puertas lógicas de entradas múltiples, como el cerebro humano. Pero, aunque fuera muy pequeño, un ordenador de ese tipo necesitaría miles de millones de cables distintos para las conexiones entre puerta y puerta. Un ordenador de este tipo sería imposible de construir en la práctica.

Con los nuevos circuitos integrados, sin embargo, la cosa cambia. No hay cables, y las conexiones –microscópicas– forman parte de la estructura misma del circuito. Esta innovación, que parece un simple progreso tecnológico, representa, sin embargo, una nueva frontera en la evolución de los ordenadores, desde el momento en que permite la construcción de máquinas con puertas lógicas de funcionamiento similar al del cerebro humano. En base a las previsiones actuales, tales ordenadores serán construidos en los años noventa. Rivalizarán en muchos aspectos con la mente humana y tendrán muchos de los atributos de la vida inteligente: capacidad de respuesta ante el mundo circundante, facultad de aprender mediante la experiencia y solícita comprensión de ideas nuevas. ¿Serán organismos vivientes?

La mayoría de la gente pensará que un ordenador no podrá ser nunca un organismo viviente porque no tiene sentimientos ni emociones; no come, no se mueve, no crece; y está hecho de metal, semiconductores y plástico, en vez de carne y hueso. Sin embargo, la mayor parte de estos atributos podrían incorporarse fácilmente a un ordenador, si se quisiera. Por ejemplo, se le pueden añadir ruedas y un motor. Y este ordenador con ruedas podrá ser programado de forma que sea capaz de dirigirse hacia una toma de corriente y conectarse a ella para tomar un «tentempié» –un trago de electricidad– en caso de que las baterías estén descargadas y los voltímetros indiquen las punzadas del hambre. Es cierto que hará falta alguien que suministre la electricidad, pero también la mayor parte de la gente recibe sus provisiones de otras manos. Naturalmente, hay, entre los seres humanos, quienes obtienen los alimentos; pero, si fuera necesario, no sería difícil construir un ordenador agresivo capaz de «husmear» la electricidad y obtenerla por sí mismo.

También los sentimientos y las emociones pueden ser incorporados a un ordenador, del mismo modo que la naturaleza los ha incorporado en las partes más antiguas del cerebro humano para facilitar la supervivencia del individuo. Samuel hizo este descubrimiento mientras intentaba lograr que su ordenador aprendiera más de prisa y se convirtiera en un estudiante mejor. La máquina jugaba bastante bien para un principiante, pero no era competitiva; en vez de forzar una rápida victoria cuando iba ganando, como habría hecho un jugador humano, se entretenía con todas las posibilidades alternativas. Samuel decidió entonces dotarla de una psicología alternante. Modificó el programa de forma que, cuando la máquina iba ganando, se volvía agresiva, escogiendo la jugada que podía llevarla a la victoria de la forma más rápida; cuando iba perdiendo, adoptaba tácticas dilatorias, eligiendo jugadas que prolongaran el juego.

(Fragmento de la obra EL TELAR ENCANTADO, de Robert Jastrow. © 1981 by Reader's Library, Inc.)

De la calculadora al ordenador: el programa

La primera máquina construida para realizar cálculos fue la calculadora, que, inicialmente de funcionamiento mecánico, sólo podía efectuar sumas y restas; posteriormente evolucionó y se volvió electromecánica, adquiriendo la capacidad de efectuar todas las operaciones.

La principal limitación de esta máquina estriba en que es necesario plantear los cálculos cada vez y no se pueden dar «instrucciones» relativas al trabajo a desarrollar.

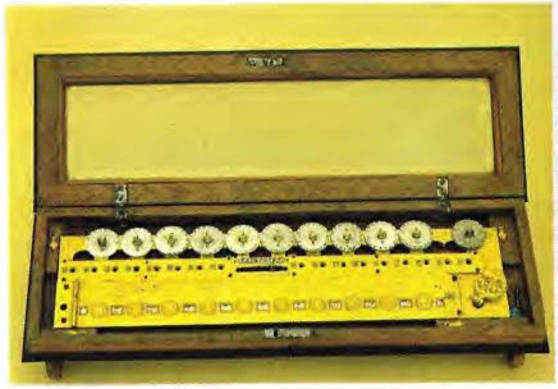
Por el contrario, los ordenadores pueden ser «instruidos», es decir, programados para realizar ciertas funciones. El ejemplo de programación que en seguida viene a la cabeza es el bien conocido de la lavadora.

Este electrodoméstico lleva incorporado un aparato que, en función del tiempo transcurrido y de la temperatura del agua, activa las distintas fases de lavado. Quien utiliza la lavadora puede elegir entre diversos tipos de lavado (más o menos largo, a temperatura más o menos elevada), es decir, puede elegir entre distintos tipos de programa.

También en este aparato existen los datos, que son el tiempo y la temperatura del agua; de la elaboración de estos datos (según el programa elegido) resultan los ciclos de lavado.

En general, el funcionamiento de una máquina programable puede esquematizarse con tres funciones:





Máquina "arithmetica" inventada en 1666 por Samuel Morland.

Pasado y presente

Los primeros instrumentos auxiliares para el cálculo –los ábacos– aparecieron en China hacia el siglo VIII a. C. Los romanos usaban también el ábaco, aunque no el de cuentas ensartadas, sino una tabla con hendiduras horizontales en las que se insertaban pequeños discos móviles. La primera hendidura representaba las unidades, la segunda las decenas, etc.

El ábaco todavía se utiliza, y hay operadores especialmente hábiles que pueden obtener con él incluso raíces cuadradas. Durante mucho tiempo éste fue el único aparato de cálculo disponible. En 1642, Pascal inventó la primera sumadora de engranajes. La sumadora de Pascal, perfeccionada, fue la primera máquina multiplicadora.

En esta máquina la operación de multiplicar se efectuaba por sumas sucesivas.

Se trataba, sin embargo, de máquinas no utilizables con fines prácticos, que no fueron más que curiosidades hasta mediados del siglo XIX. En esta época se construyeron sumadoras de funcionamiento más simple y seguro y se lograron las primeras aplicaciones prácticas.

A finales del siglo XIX (1889) se construyó la primera máquina capaz de dividir. Contenía una tabla pitagórica y efectuaba las divisiones seleccionando los resultados a partir del contenido de las tablas, pero nunca fue utilizada con fines prácticos.

Todas estas máquinas, aunque constituían los fundamentos del cálculo mecánico, eran poco utilizables, puesto que efectuaban una sola operación cada vez y no poseían memoria.

La primera máquina con memoria y con posibili-

dades de efectuar operaciones múltiples fue ideada en Inglaterra, en 1822, por el matemático Charles Babbage.
Estas nuevas posibilidades convirtieron las calculadoras en algo realmente utilizable con fines prácticos y, en base al principio de funcionamiento ideado por Babbage, se construyeron máquinas más evolucionadas, dotadas incluso de impresora. En 1929 apareció, en Estados Unidos, la primera máquina aplicable al cálculo científico.
A lo largo de esta evolución los dispositivos seguían siendo de naturaleza mecánica, y por tanto, extremadamente lentos y engorrosos. El siguiente paso de la evolución consistió en introducir mecanismos electromagnéticos movidos mediante dispositivos eléctricos. La innovación llevó a la construcción, en 1944, del MARK I, una máquina que, para su época, tenía una potencia enorme: 10 operaciones por segundo. Sus dimensiones eran notables: 18 metros de longitud y 2,5 de altura. Apenas tres años después, en 1947, se construyó la primera máquina con válvulas electrónicas. Las operaciones se efectuaban mediante circuitos eléctricos y no por medios mecánicos, aunque el principio del funcionamiento seguía siendo el que ideara Babbage.
Hasta entonces no se había introducido el concepto de programa; efectivamente, el operador debía teclear, en cada momento, el cálculo a efectuar.
En 1950 apareció la noción de programa: se suministraban a la máquina todas las indicaciones relativas a los cálculos a efectuar y, acto seguido, la máquina los realizaba sin más intervención humana. Había nacido el ordenador.
En los años siguientes este sector ha tenido un desarrollo vertiginoso, sin igual en ninguna otra ciencia.
Las válvulas electrónicas fueron sustituidas por los *transistores*, capaces de realizar las mismas funciones con un tamaño decenas de veces menor y con escasa dispersión de calor. El paso siguiente fue la introducción de los *circuitos integrados*. Esta nueva tecnología ha permitido construir ordenadores de dimensiones muy reducidas y con gran velocidad de cálculo.
En los modernos ordenadores se puede obtener 1.000.000 de operaciones por segundo en lugar de las 10 del MARK I.
La siguiente tabla resume las principales etapas de este desarrollo:

siglo VIII a.C.	Ábaco
1642	Primera sumadora
1850	Primeras aplicaciones prácticas de las sumadoras
1889	Se consigue efectuar divisiones con una máquina
1929	Cálculo científico, impresora
1944	MARK I - Dispositivos electromagnéticos
1950	Programación
1955	Transistores
Años 70	Circuitos integrados
Años 80	Integración a gran escala

Moderno centro de cálculo que utiliza un ordenador de alta potencia.



Interna

El tejedor de números: historia de un descubrimiento

Charles Babbage, famoso inventor y matemático, considerado como el inventor del ordenador (si es posible remontarse al origen de un invento), nació en Inglaterra en 1792. Como muchos sabios de la época, heredó una fortuna considerable y la gastó en la empresa loca y ambiciosa que constituyó el trabajo de toda su vida. A principios del siglo XIX empezaba a notarse la progresiva complejidad de la existencia; eran cada vez más las personas que vivían dedicándose exclusivmente a los cálculos y a la recopilación de datos de todo tipo. Una de las tareas más arduas y susceptibles de error era la recopilación de las tablas logarítmicas; Babbage continuamente encontraba en ellas errores banales. Pensando precisamente en la exasperante índole repetitiva de estos cálculos y en la consiguiente pérdida de tiempo, Babbage empezó a concebir la idea de que una máquina podría realizar ese tipo de operaciones con mucha más facilidad y exactitud. Pronto se formó en su mente el esquema básico para la construcción de una máquina de este tipo; en 1821 se sintió lo suficientemente seguro de su proyecto como para anunciar a la Royal Astronomical Society que iba a construir un prototipo y a hacerles una demostración. Explicó que su máquina funcionaría según el «método de las diferencias» y dio detalles bastante convincentes sobre su funcionamiento. No nos detendremos a explicar cómo operaba la máquina; baste decir que podía resolver ecuaciones polinómicas calculando diferencias sucesivas entre conjuntos de números. La mostró a una asamblea de la Sociedad en 1822, y la presentación tuvo tan buena acogida que la medalla de oro de la institución fue concedida a su informe Observaciones sobre la aplicación de las máquinas al cálculo de las tablas matemáticas. *Fue la primera medalla de oro. Entusiasmado, Babbage empezó a construir la versión final de la máquina. Incluso una ojeada superficial a los planos de Babbage revela hasta qué punto eran ambiciosos. Los principios fundamentales, los engranajes y los mecanismos no se diferenciaban en nada de los construidos hasta entonces, pero la máquina era extremadamente grande y compleja. Además, con su optimismo sin límites, Babbage pretendía que la máquina no sólo calculara las tablas, sino que las imprimiera sobre papel. Pidió una subvención al gobierno británico, que le concedió 1.500 libras. Se construyó un taller en los terrenos de los Babbage, se contrataron hábiles operarios y finalmente comenzaron los trabajos.*

La principal tarea consistía en fabricar con tornos especiales y con la mayor perfección posible los centenares de ejes, ruedecillas y engranajes necesarios para construir las partes móviles de la máquina diferencial. Y aquí aparecieron los primeros obstáculos serios. La pequeña unidad construida para la Astronomical Society era sólo un prototipo: las mínimas irregularidades de sus componentes podían crear inestabilidades en el sistema, pero no hasta el punto de bloquear su funcionamiento global.

Pero en la máquina diferencial cualquier serie de pequeñísimas imperfecciones tendía a acumularse provocando grandes vibraciones y bloqueos.

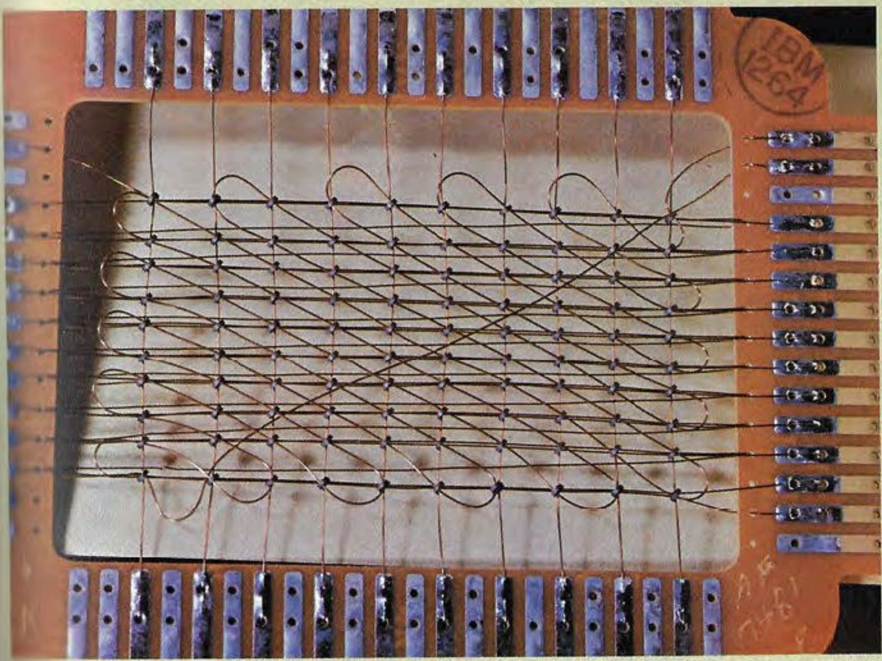
Babbage redujo las tolerancias de elaboración e incitó a sus operarios a trabajar con más precisión. Hubo una mejora, pero no lo suficientemente adecuada a la complejidad global del sistema. Insensible a los primeros fracasos, tal vez por testarudez, Babbage presionó a sus mecánicos en el vano intento de obtener resultados mejores de lo que los instrumentos y los materiales de la época podían permitir. Pidió más dinero al gobierno. Le dieron hasta 17.000 libras; luego decidieron que ya era suficiente y el proyecto fue suspendido en 1833.

Si Babbage hubiera sido una persona razonable, cosa que no era, ese hubiera sido el momento de detenerse a contemplar las toneladas de engranajes de latón y de plomo y aceptar el hecho de que pretendía adelantarse cien años a su época. Por el contrario, su mente infatigable empezó a dedicarse a un esquema aún más ambicioso. En este momento nació el concepto de ordenador.

Si bien la máquina diferencial constituía un paso adelante con respecto a cualquier otra cosa construida anteriormente, era capaz de realizar una única tarea: resolver ecuaciones polinómicas. El resultado se obtenía mediante una serie de movimientos preestablecidos: un sistema de engranajes hacía girar otro, una palanca movía otra y así sucesivamente, en una secuencia previsible. En otras palabras, el sistema era lo que hoy llamaríamos un «ordenador dedicado». Estaba en condiciones de realizar una sola función, para la cual había sido creado.

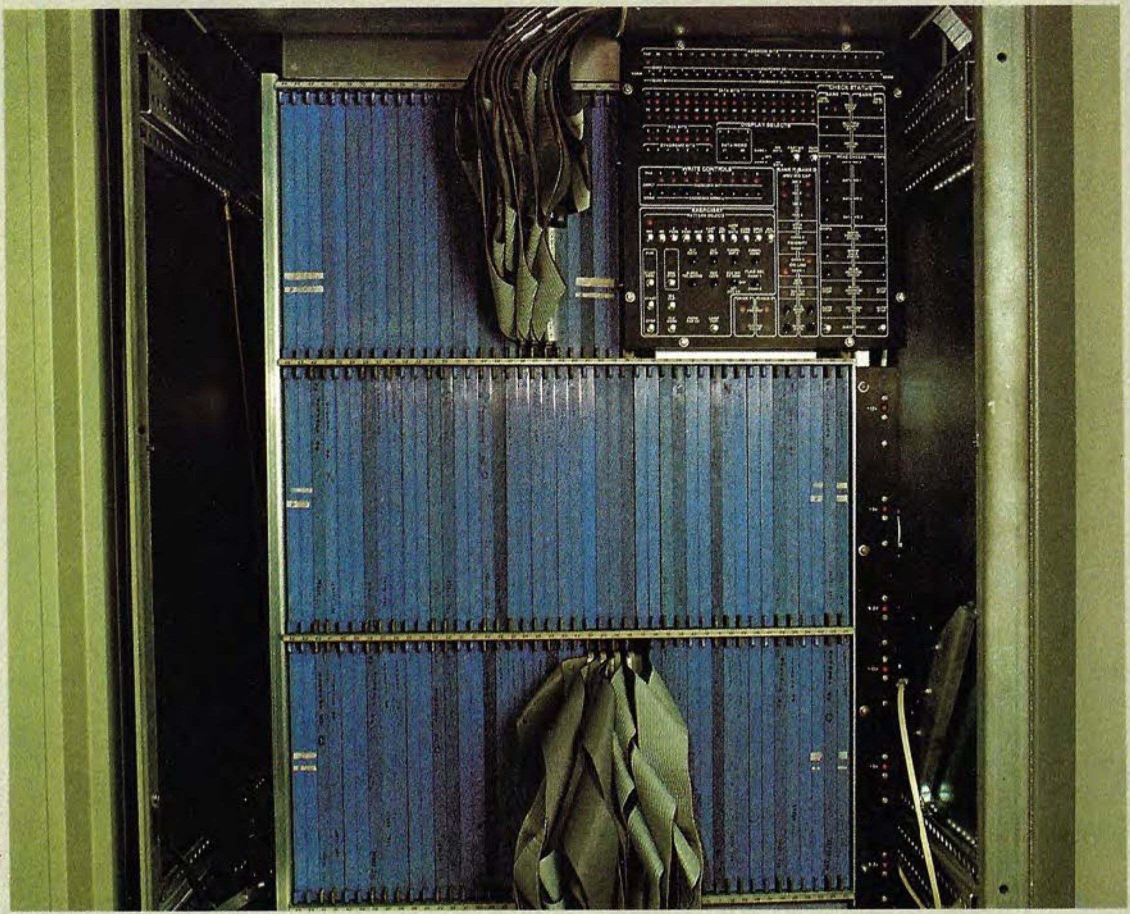
Reflexionando, Babbage se dio cuenta de que estaba siguiendo un camino equivocado. Una máquina que podía realizar un determinado tipo de cálculos, podía, seguramente, realizar cualquier tipo de cálculos. Esta idea fue demostrada matemáticamente, casi un siglo después, por otro genio inglés: Alan Turing. De su demostración surgió un concepto revolucionario y estimulante. ¿Por qué no construir una máquina sin una tarea preestablecida, que pudiera realizar toda una serie de tareas como y cuando su usuario quisiera? Tal vez fuera mucho más complicada que la máquina diferencial y de construcción mucho más costosa, pero también resultaría mucho más útil. Podría haber parecido «obvio» que la nueva máquina sería un conjunto de otras muchas máquinas –la parte «a» dedicada a realizar la tarea «a», la parte «b» dedicada a la tarea «b», etc.–, todas ellas con un mismo mecanismo de entrada y una conexión común a la impresora u otro dispositivo de salida. Pero Babbage descartó lo «obvio» y llegó a su extraordinaria intuición. El planteamiento de la máquina debía permitir que sus mecanismos se pudieran usar de muchas formas distintas. De este modo, para realizar una determinada tarea se haría funcionar una secuencia específica de actividades internas, y cada tarea iría ligada a modelos únicos de funcionamiento de los mecanismos. Sólo faltaba una idea brillante para encontrar la forma de «decirle» a la máquina qué acción, de entre la enorme variedad de acciones posibles, debía emprender en cada momento. Babbage llamó a este dispositivo «máquina analítica»; conviene subrayar que estaba hablando propiamente de un «ordenador programable».

En primer lugar, había un conjunto de dispositivos de entrada, que permitían introducir números o instrucciones en el interior de la máquina. Luego había una unidad aritmética o procesador, es decir, la parte de la máquina que realmente efectuaba los cálculos. Babbage la llamó «la muela». En tercer lugar, había una unidad de control que garantizaba que la calculadora realizara la tarea requerida y no otra, y efectuara todos los cálculos en la secuencia correcta. Había un almacén, o memoria, que retenía las cifras en espera de su turno para ser elaboradas. Finalmente, había un dispositivo de salida propiamente dicho. Éstas, aunque de forma muy esquemática, son las cinco partes fundamentales de cualquier ordenador, antiguo o moderno. Para sus dispositivos de cálculo y memoria, Babbage utilizó columnas y más columnas de engranajes de diez dientes, unidas por una increíble maraña de juntas y ejes de transmisión. El prototipo de la máquina diferencial «Engine» se movía a mano. Accionando una palanca, se ponían en movimiento los engranajes en el inte-



Archivo Curco/IBM

La primera forma de memoria electrónica, constituida por núcleos magnéticos, donde el elemento de memoria es un arito de material ferromagnético y el entrelazado de los cables sirve para determinar sus coordenadas (es decir, su dirección) y para leer y escribir el dato mediante impulsos eléctricos.


Dominique Fradin

Las memorias de los ordenadores evolucionan: del tipo "de núcleos" se pasó al "de semiconductores", más pequeñas y rápidas. Interior del "banco de memoria" de un ordenador moderno.

rior de la máquina, y al completarse los cálculos deseados sonaba una campana. Entonces se introducían nuevos datos, se accionaba otra vez la palanca hasta que sonaba la campana, y así sucesivamente. Pero Babbage veía claramente que era absurdo recurrir a la fuerza muscular para mover su máquina, y pensó en utilizar un motor a vapor para su última versión, la máquina analítica.

Hay algo patético y casi ridículo en el hecho de asociar la ruidosa e incipiente potencia de un motor a vapor con la delicada tarea de efectuar cálculos numéricos; pero se trata, simplemente, de otra prueba de que Babbage estaba soñando, de forma magnífica, desde luego, pero siempre soñando y anticipándose a su tiempo en varias décadas.

Para insertar y programar las instrucciones de la unidad de control, Babbage se inspiró en un invento del francés Joseph Jacquard. Jacquard había observado que los tejedores, al manejar sus telares, llevaban a cabo una tarea delicada pero esencialmente repetitiva, y que, por tanto, debía ser viable la automatización del proceso. Inventó entonces una cartulina rígida perforada. Durante el tejido, una serie de guías mueven los hilos en el telar; la función de la cartulina consistía en bloquear algunas de esas guías y permitir a las otras, que pasaban a través de los orificios, realizar el trenzado. A cada golpe de lanzadera, una cartulina con una determinada serie de orificios se interponía en el camino de las guías, controlando el diseño del tejido. Esto constituía, en esencia, un programa de control del telar, y Babbage comprendió que el método podía ser igualmente eficaz para controlar la secuencia de los cálculos en su máquina.

El paralelismo con la operación de tejido fue descubierto también por Ada, condesa de Lovelace, que, en un comentario sobre la máquina de Babbage, escribió: «La máquina analítica teje conjuntos algebraicos del mismo modo que

el telar de Jacquard teje flores y hojas». Esta inquietante mujer estaba excepcionalmente dotada para las matemáticas, además de poseer una belleza fuera de lo común. Un apunte de la época la muestra un tanto parecida a Elizabeth Taylor y con unas facciones que recuerdan a las de su apuesto padre, lord Byron. Cuando conoció a Babbage y se dio cuenta de la importancia de su proyecto, se puso a estudiar a fondo los planos de la máquina analítica y llenó las lagunas con las explicaciones orales del propio inventor. El dinero y el tiempo estaban de su parte, puesto que tenía veinte años y era rica. Pese a ello pasaron varios años antes de que llegara al fondo de la cuestión. Finalmente, publicó una larga serie de notas tituladas Comentarios sobre la máquina analítica del señor Babbage. *Constituyen una excelente lectura para quien desee estudiar el proyecto con detalle, y demuestran que lady Lovelace conocía algunos de los problemas filosóficos que se plantean al construir una máquina de este tipo y que no han sido resueltos todavía. Por ejemplo, con respecto al problema de si la máquina debía ser considerada creativa o no, escribe: «La máquina analítica no tiene la pretensión de crear nada. Puede hacer cualquier cosa que estemos en condiciones de programar. Puede efectuar análisis. Pero no puede anticipar verdades o relaciones analíticas. Su objetivo es volver accesible lo que ya sabemos». Fue un comentario muy inteligente; constituyó, al parecer, la primera declaración contundente relativa a la discusión que se vuelve a plantear cada vez que se habla del potencial intelectual de un ordenador:* un ordenador sólo puede hacer aquello para lo que está programado. *Como se verá, es un argumento tentador, aunque válido sólo superficialmente. El mérito de haberlo formulado por primera vez corresponde a lady Lovelace. Y la importancia de sus notas estriba en la repercusión que tuvieron en el propio Babbage. Hasta ese momento, y exceptuando su éxito con la Astronomical Society, Babbage nunca se había encontrado con alguien que aprobara sus trabajos, por no hablar de alguien que los entendiera. Lo que le produjo mayor satisfacción fue que la condesa se hubiera tomado la molestia de estudiar sus teorías desde el punto de vista matemático y no hubiera encontrado pegas. Él sabía que, en principio, la máquina analítica tenía que funcionar; ahora también ella lo creía. Sólo faltaba construir el malhadado aparato.*

Mientras tanto, empero, en Inglaterra había cambiado el gobierno y el ministerio competente había renunciado a financiar un proyecto que consideraba ya irrealizable. El tiempo pasaba, la máquina diferencial no era más que un conjunto incompleto de ejes y engranajes, y la máquina analítica era sólo una serie de bocetos sobre papel y un montón de notas de lady Lovelace. La decadencia se fue acelerando. Ada murió a los treinta y seis años y Babbage continuó solo, con escasos resultados. Los gobiernos iban cambiando y ninguno simpatizaba con su idea. Disraeli escribió incluso sarcásticamente que, según él, la única aplicación posible de la máquina diferencial sería calcular la enorme cantidad de dinero desperdiciada para construirla. Pero en ese mismo momento otros matemáticos e ingenieros leían con gran interés las publicaciones de Babbage y las notas de lady Lovelace. Uno de ellos, un ingeniero sueco llamado George Scheutz, empezó a construir una versión suya de la máquina diferencial. Al contrario que Babbage, la sacó adelante y obtuvo un éxito tan grande que construyó un prototipo que expuso como «primer modelo para la producción» en una exposición de ingeniería, en 1885. Entre la multitud que se agolpaba alrededor del curioso pero tangible dispositivo estaba también Babbage. Cuando le pidieron un comentario sobre la máquina de Scheutz, fue muy amable y lo felicitó, pero no es fácil imaginar qué pensamientos cruzaron por su mente.

Babbage falleció en 1871, a los ochenta y tres años. Es un hecho triste pero cierto que murió desilusionado, pese a haber concebido un proyecto tan sugestivo y revolucionario que un día cambiaría el mundo. Sus contemporáneos lo consideraron un genio iluso y descabellado, pero sin duda un genio. Estaban todos tan convencidos de su extraordinaria inteligencia que, tras su muerte, su cerebro fue sometido a un cuidadoso examen para ver si presentaba características físicas que lo diferenciaran de un cerebro común. Uno de los más famosos cirujanos de la época, sir Victor Horsely, se ocupó del examen y anunció que aquel cerebro no le parecía diferente de otros muchos que había examinado. Sin embargo, no se atrevieron a tirarlo; todavía se conserva en el Museo Hunteriano del Real Colegio de Cirujanos, donde lo muestran a quien desee verlo.

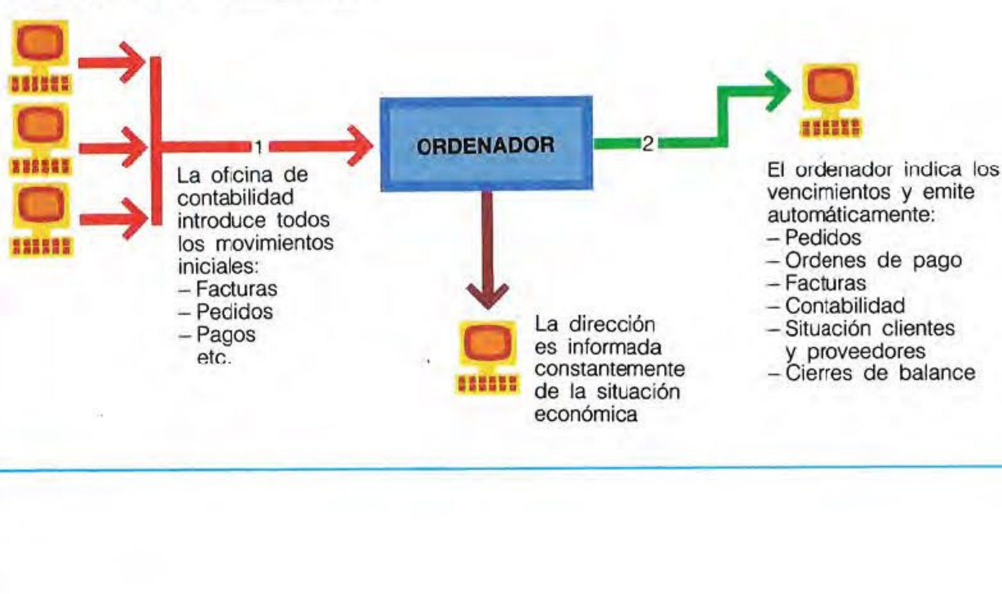
Aplicaciones y evolución de los ordenadores

No es posible identificar, en los últimos años, hitos concretos, pues se ha venido produciendo un continuo desarrollo, y los perfeccionamientos y adelantos se han sucedido sin solución de continuidad en cuestión de meses o días. Este fenómeno está ligado al profundo cambio social; efectivamente, la necesidad primordial en un mundo industrializado es la información. El rápido intercambio de informaciones puede significar el éxito de una investigación científica, la elección correcta en una decisión económica o, simplemente, un ahorro de tiempo.
El ordenador es el instrumento ideal para estas aplicaciones; su capacidad de memorización almacena grandes cantidades de información a la que el usuario tiene acceso, para seleccionar y extraer datos en un tiempo brevísimo.
Los ejemplos son interminables: piénsese en la gestión de una biblioteca. La búsqueda de los volúmenes sobre un determinado tema, si la realiza un hombre, es tediosa y requiere un tiempo considerable; si, por el contrario, la biblioteca es gestionada por ordenador, se puede obtener la información en un tiempo muy breve y sin que el usuario tenga que dedicarse a una actividad manual que roba tiempo a la creativa.
También en las empresas, las distintas secciones han de conocer la situación general para poder programar correctamente sus respectivas actividades. Una sección de expedición de piezas acabadas, por ejemplo, ha de saber cuántos objetos ha terminado producción; a su vez, la sección de producción ha de conocer la cantidad de componentes que hay en el almacén. Este complejo bloque de información sólo puede ser gestionado eficazmente mediante un ordenador. (Ver en los gráficos los posibles usos industriales de un ordenador).
La ciencia que estudia las formas y maneras de poner la información a disposición de los usuarios es la **informática**.

PRINCIPALES CAMPOS DE APLICACION DEL ORDENADOR EN UNA EMPRESA



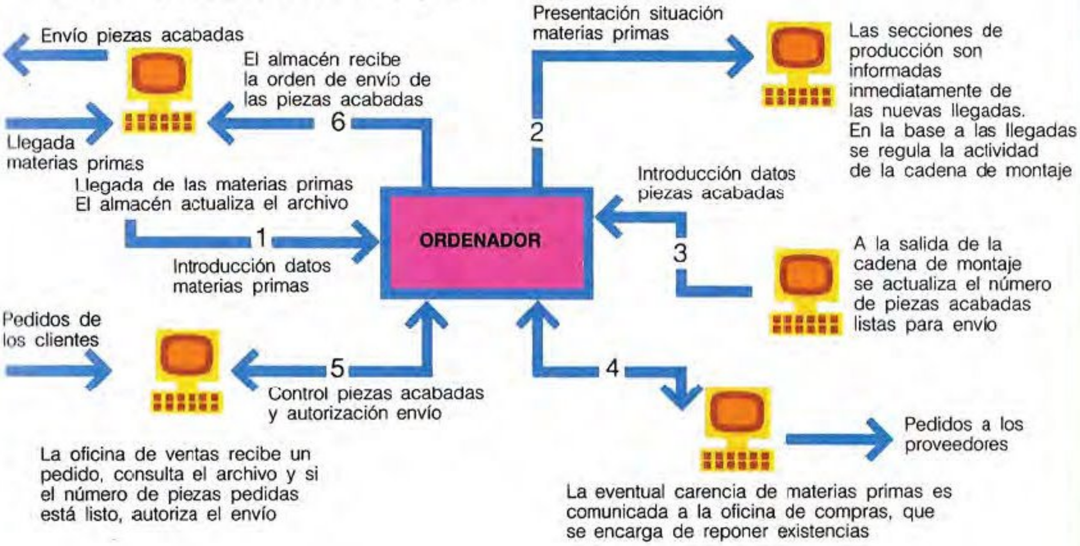
Gestión de la contabilidad



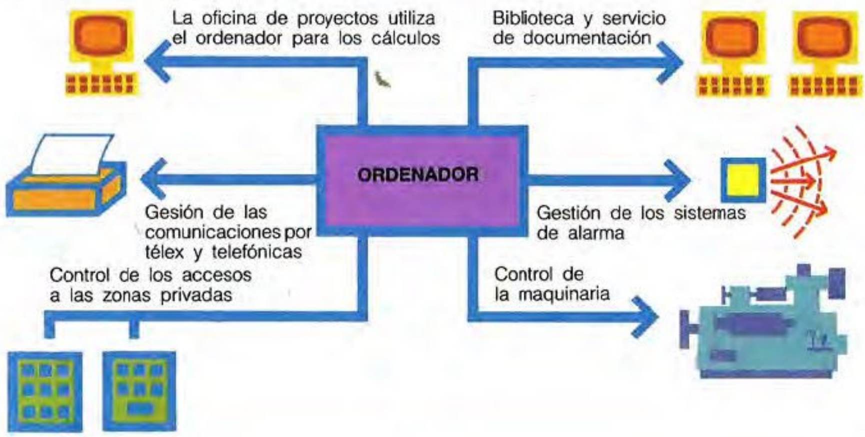
Gestión del personal



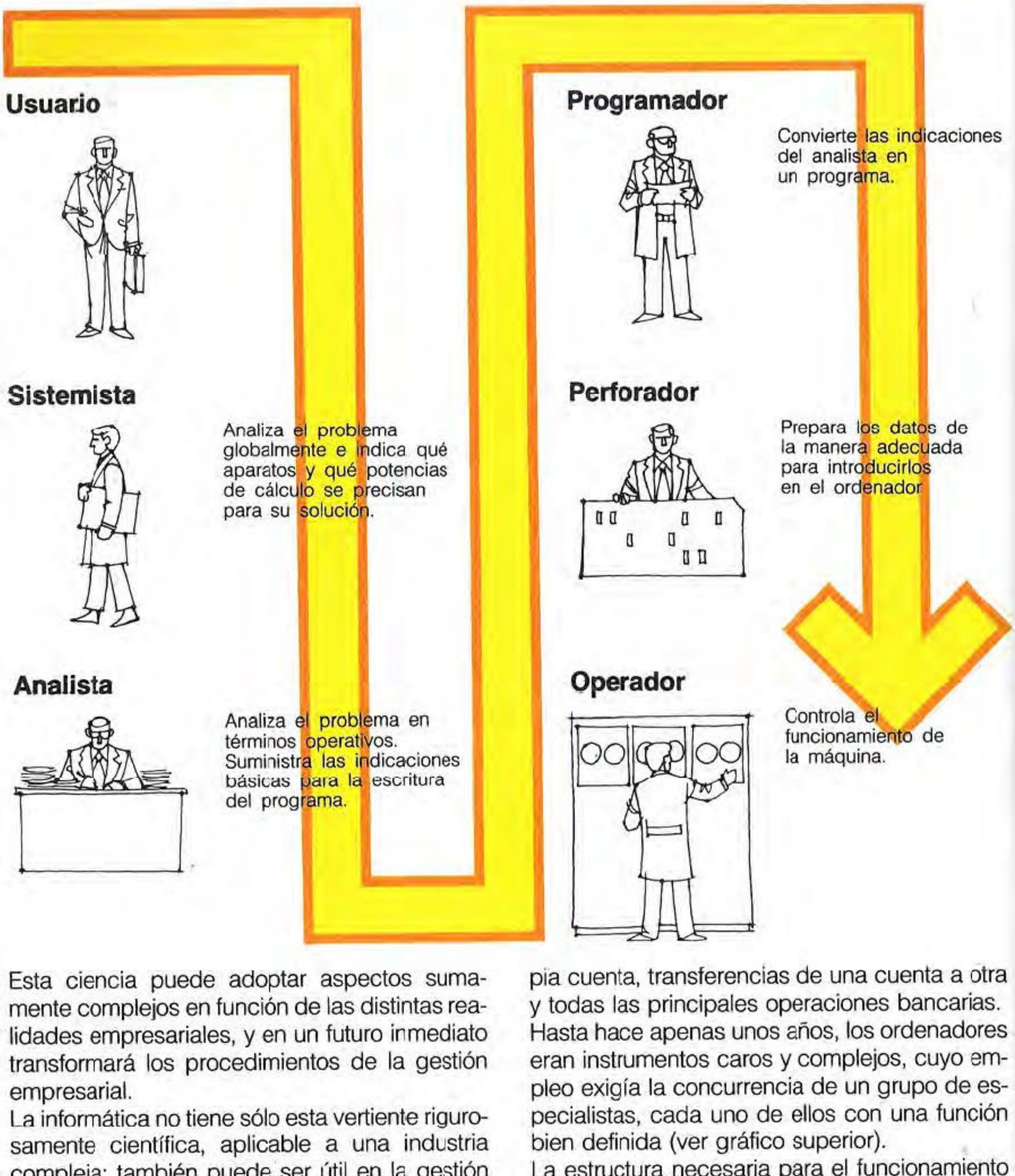
Sistema informativo para la producción



Aplicaciones especiales



ESTRUCTURA DE UN CENTRO DE CALCULO



Esta ciencia puede adoptar aspectos sumamente complejos en función de las distintas realidades empresariales, y en un futuro inmediato transformará los procedimientos de la gestión empresarial.

La informática no tiene sólo esta vertiente rigurosamente científica, aplicable a una industria compleja; también puede ser útil en la gestión familiar.

¿Cuántos conocen, en todo momento, el saldo de su cuenta corriente? La mayoría maneja mentalmente una cifra vagamente aproximada y a veces alejada de la realidad. Este problema está a punto de ser resuelto y existen ya servicios bancarios a domicilio. Mediante un terminal, se pueden efectuar consultas sobre la propia cuenta, transferencias de una cuenta a otra y todas las principales operaciones bancarias.

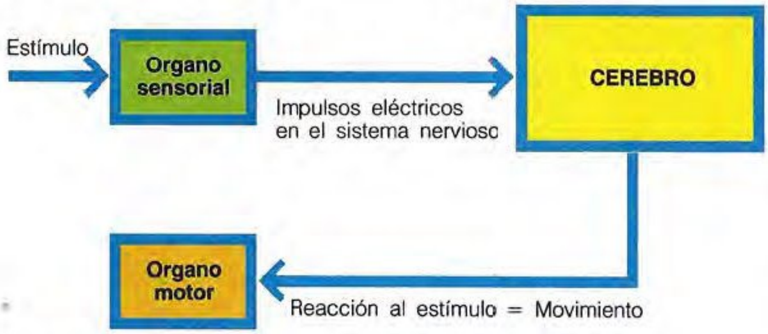
Hasta hace apenas unos años, los ordenadores eran instrumentos caros y complejos, cuyo empleo exigía la concurrencia de un grupo de especialistas, cada uno de ellos con una función bien definida (ver gráfico superior).

La estructura necesaria para el funcionamiento de estas máquinas implicaba un ingente desembolso económico y, por tanto, los ordenadores sólo podían ser utilizados en propiedad por grandes industrias; con gastos menores, pero aun así considerables, se podía alquilarlos por breves períodos en centros de servicio.

En los últimos años, sin embargo, el desarrollo tecnológico ha hecho posible la traducción de

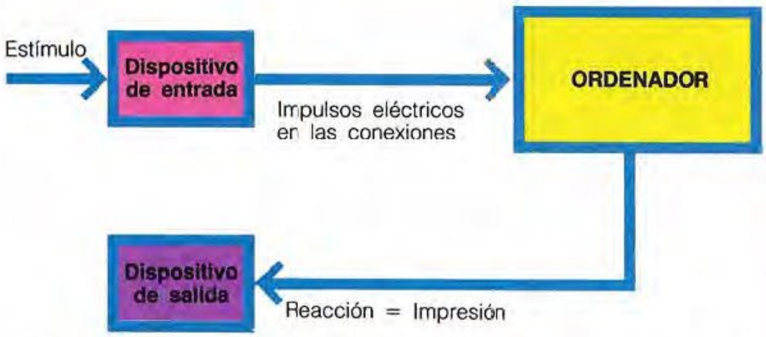
ESQUEMA LOGICO DEL PROCESO ACCION-REACCION

Proceso acción-reacción en el cerebro



Un estímulo, por ejemplo, la picadura de un insecto, produce una serie de impulsos eléctricos. El cerebro los elabora y ordena, como reacción, una acción motriz.

Proceso acción-reacción en el ordenador



En el ordenador, el estímulo es la orden de efectuar un cálculo determinado. Esta orden se convierte en señales eléctricas en el dispositivo de entrada y se envía, mediante las adecuadas conexiones, al ordenador. Las señales son interpretadas y se genera la reacción, es decir, la impresión del resultado.

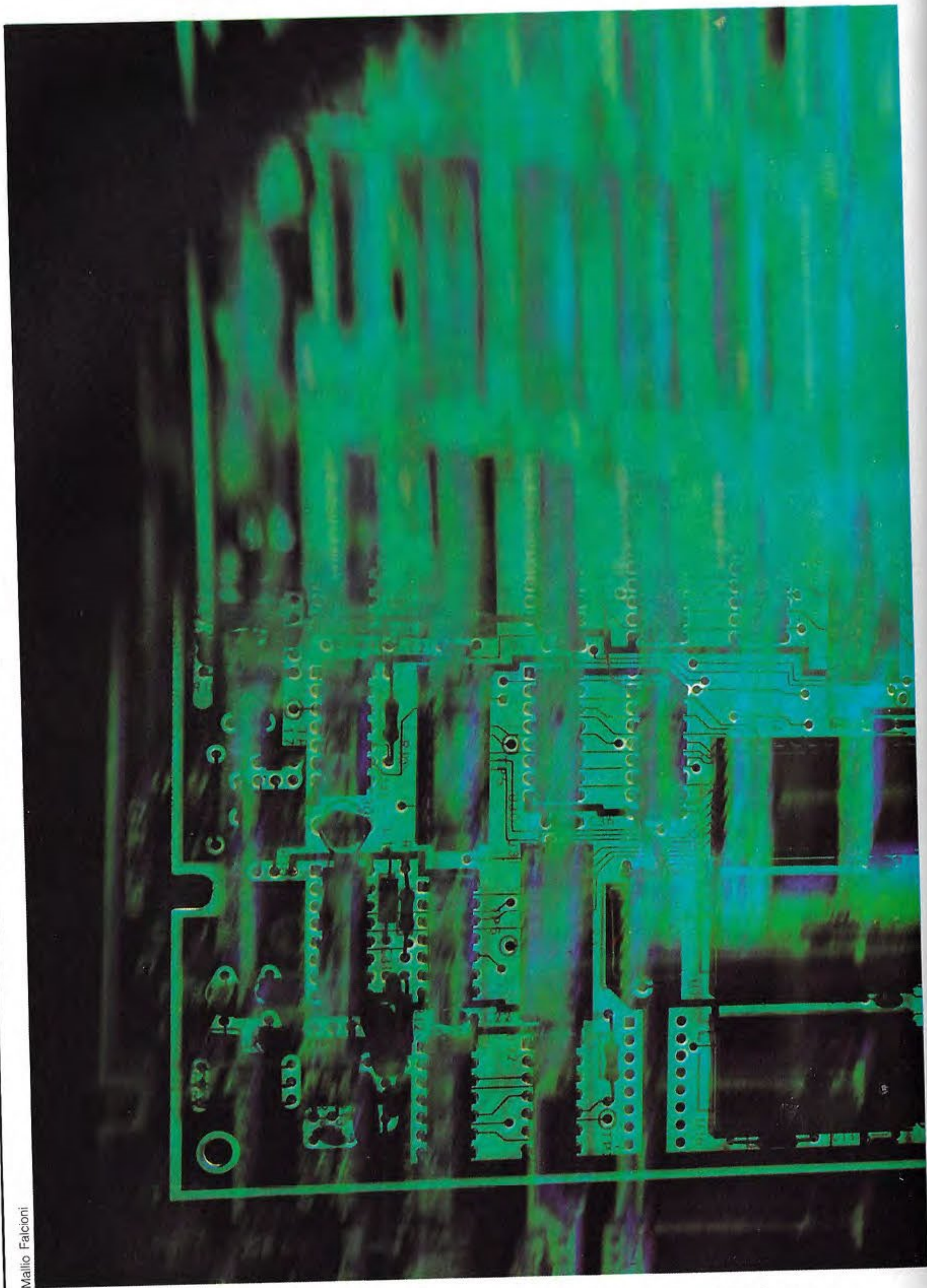
ordenadores de dimensiones muy reducidas, fáciles de usar y de bajo costo.

Es fácilmente previsible cuál puede ser su desarrollo futuro; actualmente los ordenadores ya hablan, pueden leer textos, hacer dibujos, jugar al ajedrez, traducir a varias lenguas, componer música; en un futuro próximo podrán, con toda seguridad, interpretar órdenes impartidas con la voz. El desarrollo más interesante se producirá en el campo de la medicina.

Existen, en efecto, notables analogías entre el mecanismo de percepción de los estímulos por parte del cuerpo humano y el método de adquisición de datos en un ordenador: la neurona (unidad fundamental del sistema nervioso) puede hallarse en dos únicos estados: excitada o no excitada. Y precisamente en esta condición de dos únicos estados posibles se basan el funcionamiento y los principios constructivos de l ordenadores digitales.

Basándose en estas analogías es, pues, posible estimular partes del funcionamiento del cerebro humano con un ordenador (ver gráfico superior, que representa el organigrama detallado del proceso acción-reacción).

El desarrollo de los ordenadores y las analogías con el cuerpo humano han propiciado el nacimiento de una nueva ciencia: la **robótica**. Los descubrimientos en este campo han producido máquinas que pueden realizar casi todos los trabajos manuales y que poseen nuestros mismos sentidos.



Maliio Falcioni

Un componente fundamental de todas las máquinas electrónicas: el circuito impreso.

La lógica del ordenador

Con el término genérico de ordenador designaremos, de ahora en adelante, un conjunto de componentes, no necesariamente eléctricos, capaces de realizar una serie de operaciones preestablecidas según una secuencia lógica. Existen, en efecto, ordenadores de fluido (por ejemplo, de aire comprimido) en los cuales, aunque conservando los principios de funcionamiento de los aparatos electrónicos, todas las funciones son realizadas por dispositivos fluidodinámicos.

Una secuencia de funciones puede ser temporal o lógica. En la serie temporal, cada función es realizada siempre después de la precedente y antes de la siguiente.

En una serie lógica, el orden cronológico normal puede modificarse en función de determinados acontecimientos. Pongamos un ejemplo:

- Un semáforo funciona según una secuencia temporal: efectivamente, los colores se suceden siempre en el mismo orden.
- El empleo del cambio de marchas de un automóvil es una secuencia lógica: el paso de un estado a otro (marchas) viene determinado por acontecimientos que no tienen una relación fija con el tiempo (ver gráfico).

El ordenador, utiliza una secuencia lógica. Todas las funciones a realizar están preestablecidas y constituyen el programa.

En el programa habrá que prever todos los posibles caminos subsiguientes a eventuales elecciones lógicas.

Los ordenadores se pueden dividir en dos tipos fundamentales:

- **Ordenadores analógicos**
- **Ordenadores digitales**

Ordenadores analógicos

El término «analógico» alude a su principio de funcionamiento: el cálculo matemático a resolver se realiza por medio de una analogía con adecuadas magnitudes físicas. La regla de cálculo, por ejemplo, es un dispositivo analógico, puesto que permite resolver operaciones aritméticas mediante analogías con longitudes (ver gráfico superior en la pág. 34).

En este tipo de máquinas el resultado de una operación se obtiene aprovechando determinadas leyes físicas. Para cada tipo de operación se necesita, por tanto, un circuito adecuado. Son máquinas destinadas a usos concretos.

Ordenadores digitales

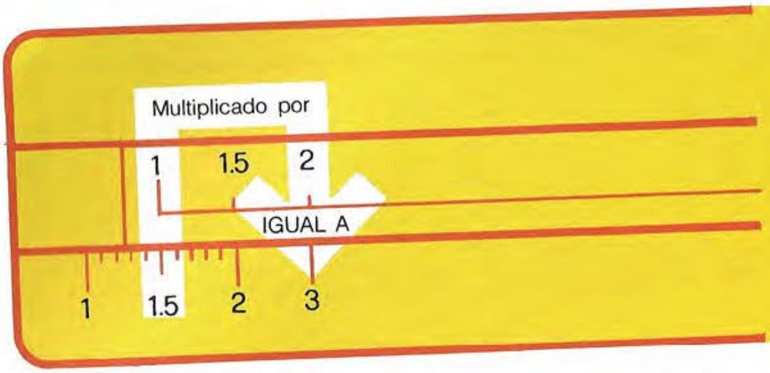
Nos referiremos sólo a los ordenadores electrónicos, aunque cuanto se diga puede aplicarse a cualquier tipo de máquina (ordenadores de fluido o mecánicos). El ordenador electrónico es una máquina en la que todas las funciones son realizadas por circuitos electrónicos.

Secuencia temporal: *el semáforo tiene una programación temporal*



Secuencia lógica: *el uso del cambio de marchas está en función de elecciones basadas en acontecimientos externos*





REGLA DE CALCULO

La regla de cálculo es un instrumento de uso común para efectuar cálculos, incluso complicados, excepto sumas y restas. Es un ejemplo de instrumento analógico: efectivamente, realiza los cálculos por analogía con las longitudes de las escalas grabadas en ella.

En general, los circuitos electrónicos pueden ser de dos clases:

- **Digitales**
- **Analógicos**

Un circuito digital sólo puede reconocer dos situaciones:

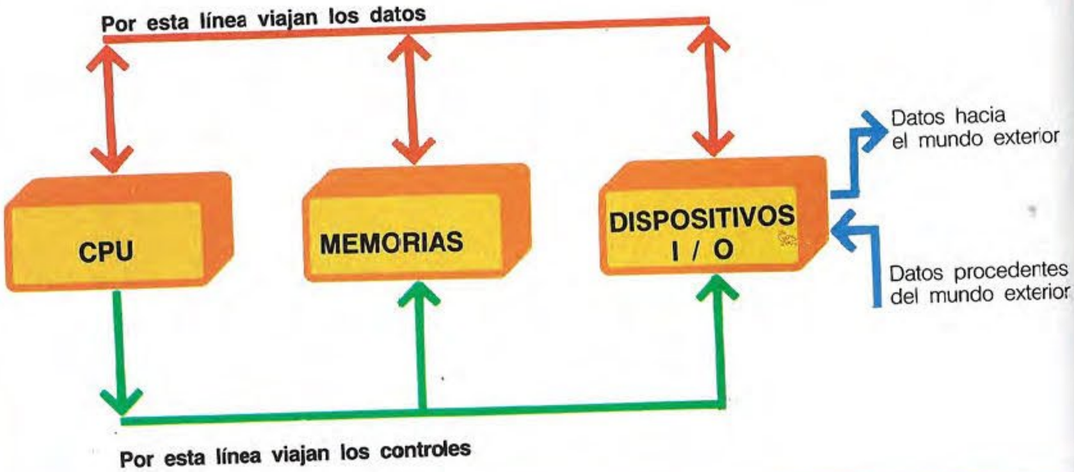
Presencia de señal = ON

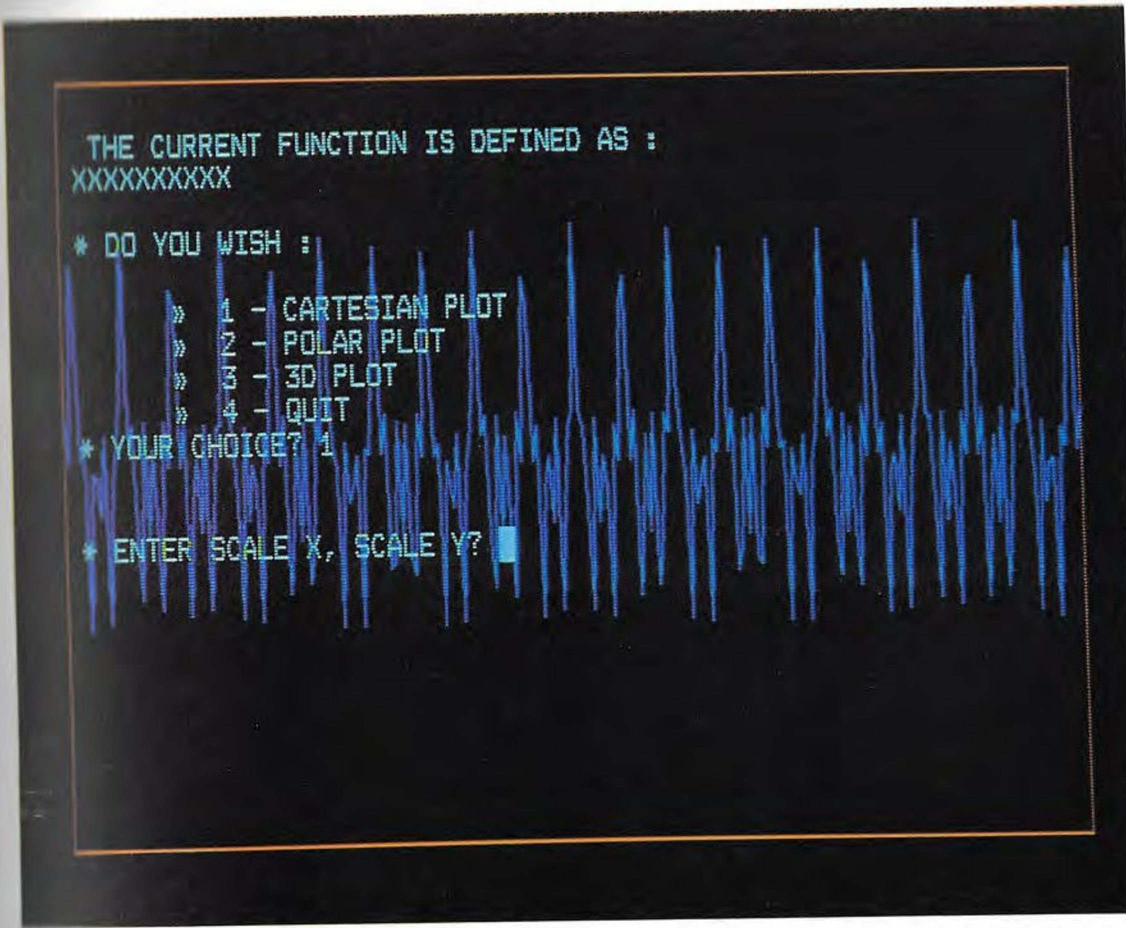
Ausencia de señal = OFF

Por el contrario, el circuito analógico puede determinar, además de la Presencia/Ausencia de señal, también su valor.
La película fotográfica es un dispositivo analógico, pues suministra un dato (la fotografía) proporcional a la señal (la luz). Un interruptor es un dispositivo digital, puesto que sólo tiene dos posibilidades: Cerrado o Abierto.
El aparato de radio, el televisor, el tocadiscos, son todos ellos aparatos analógicos, puesto que suministran sonidos o imágenes proporcionales a las señales recibidas.

Un ordenador digital es una máquina capaz de realizar funciones lógicas mediante una serie de estados ON/OFF (Presencia/Ausencia)

ESTRUCTURA MINIMA DE UN SISTEMA DE MICROORDENADOR





Centro THC

Una modalidad de display: la pantalla de vídeo. Este tipo de display es el más evolucionado; hay otros que sólo pueden mostrar una línea cada vez.

Las funciones que puede realizar un ordenador son:

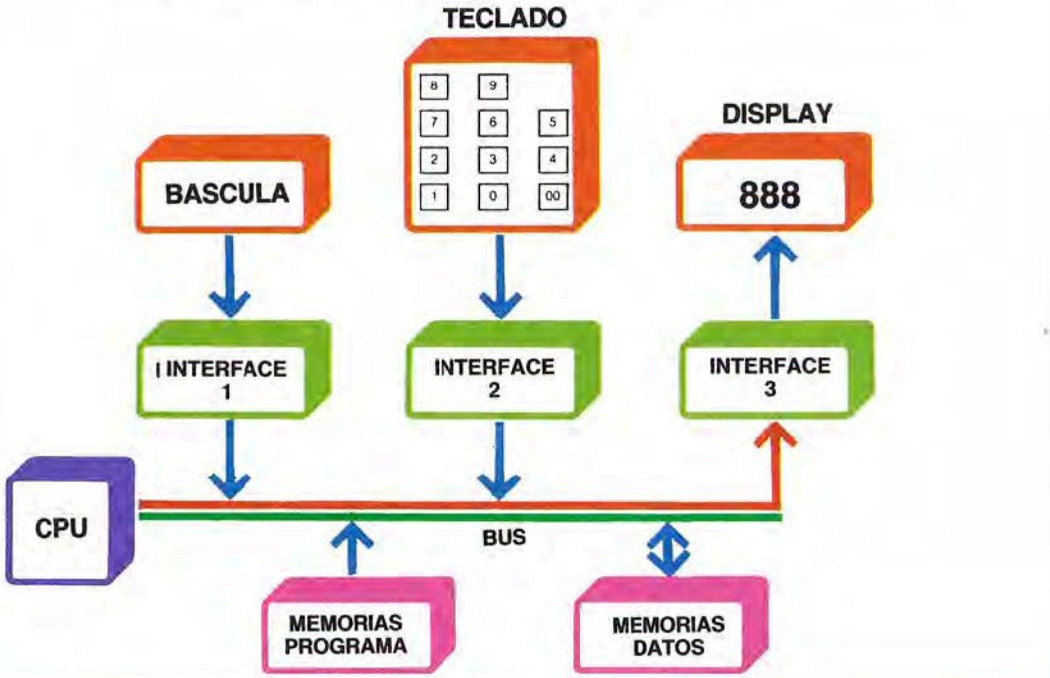
- Operaciones aritméticas entre dos datos.
- Comparación entre dos datos y consiguiente elección.
- Transferencia y memorización de datos.

A pesar de este limitadísimo número de funciones principales, es posible, mediante adecuadas combinaciones, efectuar cualquier cálculo.
De las funciones mismas se puede deducir el esquema de un ordenador.
Deberá contener:

- Un dispositivo capaz de efectuar las operaciones aritméticas y de comparación (UNIDAD CENTRAL DE PROCESO = CPU).
- Un dispositivo (o más de uno) capaz de memorizar los datos (MEMORIAS).
- Un dispositivo capaz de comunicar los resultados al usuario y de aceptar datos del exterior (dispositivos de ENTRADA/SALIDA = Input/Output = I/O).

El esquema de un sistema se muestra en el gráfico inferior de la página contigua.
La unidad central (CPU) se encarga de efectuar los cálculos y enviar las órdenes que activan las otras unidades.
La memoria es un dispositivo capaz de asimilar un dato, conservarlo y devolverlo cuando sea requerido.
Los dispositivos I/O se encargan de transformar los datos de forma que sean intercambiables entre el mundo exterior y la máquina.
Para entender el funcionamiento de un sistema de ordenador, véase el ejemplo expuesto en el gráfico de la pág. 36, que se refiere a una báscula electrónica de las utilizadas comúnmente en los comercios. La línea de transmisión de datos y la de control se representan con una sola conexión denominada BUS.

ESQUEMA DE UNA APLICACION: BASCULA ELECTRONICA



Las funciones que ha de realizar el sistema son:

A / Pesada de un objeto y determinación del valor del peso

La pesada es efectuada por la báscula. Este dato es transformado en información digital en la interface 1.

La interface 1 es un dispositivo de Entrada.

B / Determinación del precio del objeto

La función es realizada por el teclado y la interface 2.

La interface 2 es otro dispositivo de Entrada.

C / Cálculo del precio total

La función es realizada por la CPU.

D / Presentación del resultado

La función es realizada por el conjunto interface 3–display*.

La interface 3 es un dispositivo de Salida (Output).

En el esquema aparecen otros dos bloques:

- **Memorias Programa**
- **Memorias Datos**

* Con el término «display» se designa cualquier dispositivo que sirva para visualizar datos. Más adelante se ampliará este concepto.

Veamos ahora por qué es necesario el uso de dos memorias.

La función de la Memoria Programa es obvia. Contiene una serie de informaciones que, adecuadamente interpretadas, indican a la CPU la secuencia (A-B-C-D) de funciones a activar.

La Memoria Datos cumple la función de almacenar los datos en el momento de su introducción y suministrarlos cuando la CPU los requiere.

La secuencia de las operaciones es:

1 / La báscula indica que ha sido puesto un objeto sobre el platillo.

2 / La CPU pide a la Memoria Programa la operación a efectuar.

3 / La Memoria Programa indica que hay que tomar el dato (peso) y transferirlo en la Memoria Datos.

4 / Una vez efectuada la transferencia, la CPU vuelve a consultar la Memoria Programa.

5 / La Memoria Programa indica que hay que tomar el dato del teclado (precio).

6 / La CPU acciona la interface 2 y toma el dato del teclado.

7 / La CPU consulta la Memoria Programa.

8 / La Memoria Programa indica que hay que tomar el dato obtenido en el paso 3 y multiplicarlo por el obtenido en el paso 5.

9 / Finalmente, tras un ulterior ciclo de consultas de la CPU a la Memoria Programa, el resultado del paso 8 es enviado a la interface 3, que lo presenta en el display.

La secuencia de las operaciones descritas, traducida en una simbología comprensible para la máquina, constituye un programa.

Dónde y cómo utilizar un ordenador

El empleo de un ordenador está supeditado a una condición esencial:

HAY QUE PODER SUMINISTRARLE TODOS LOS DATOS NECESARIOS

Esta afirmación no es tan superflua como parece. En los contactos con los usuarios, es frecuente escuchar demandas cuando menos extrañas, puesto que quienes no conocen estas máquinas tienden a atribuirles posibilidades inverosímiles.

Hay que recordar siempre que un ordenador sólo puede suministrar elaboraciones de datos que le han sido suministrados.

El segundo punto a analizar es:

CUÁN GRANDE PUEDE SER SU UTILIDAD

No se puede asegurar, a priori, que la instalación de un ordenador sea siempre rentable.

En general, el uso de la máquina es económicamente ventajoso cuando se verifican todas o parte de las siguientes condiciones:

1 / **Los datos están todos disponibles: no es necesario inventar nada.**

2 / **Hay que elaborar una gran cantidad de datos de forma repetitiva.**

3 / **Hay que realizar una gran cantidad de cálculos.**

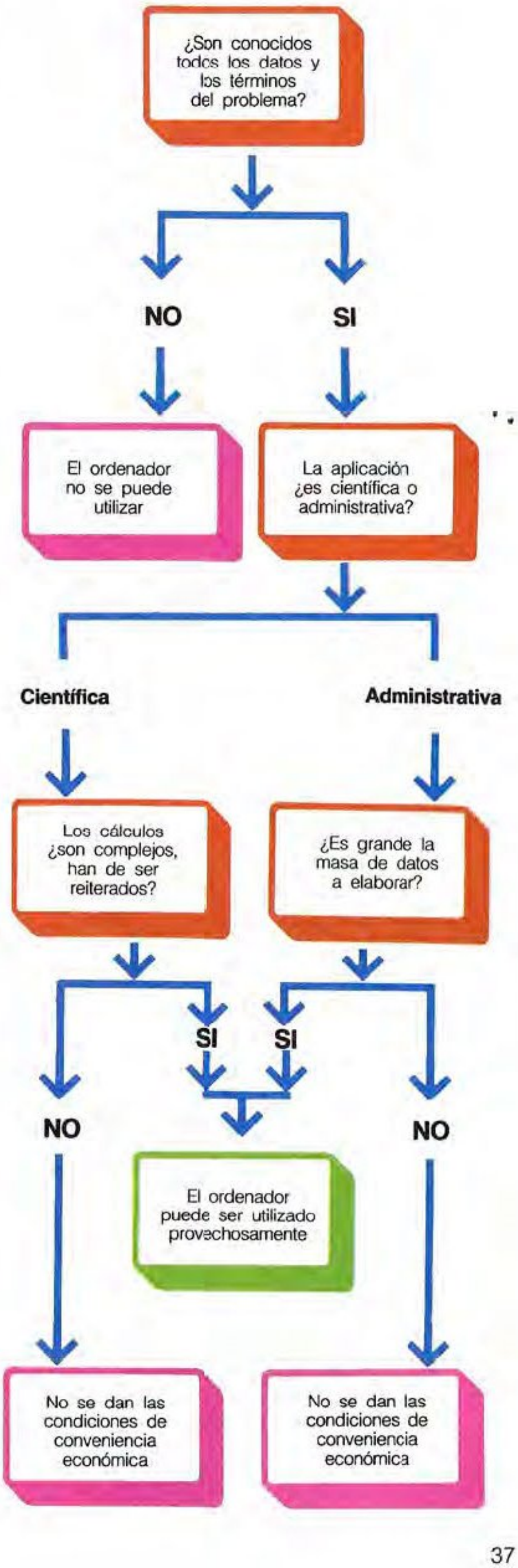
4 / **Hay que controlar constantemente un fenómeno determinado.**

El punto 1 es la condición básica.

El ordenador no puede suministrar más que aquello para lo que ha sido programado, es decir, no puede dar resultados si el programador no conoce exactamente el método a seguir.

El punto 2 indica cuán útil puede ser un ordenador. Al instalar este tipo de máquinas hay que tener en cuenta que, además del desembolso inicial, hay que desarrollar el software (programas). Como veremos más adelante, el desarrollo del software requiere unos gastos notables, que sólo pueden ser compensados si se obtiene una clara agilización del trabajo.

ANALISIS DEL PROBLEMA



El punto 3 atañe especialmente a los problemas científicos. En muchos casos, en el desarrollo de un proyecto, es necesario repetir los cálculos para buscar la solución óptima mediante aproximaciones sucesivas.
El ultimo punto (4) se refiere a la posibilidad de utilizar un ordenador para controlar y dirigir un proceso cualquiera. En este tipo de aplicación el ordenador puede ser casi siempre una inversión rentable.
En el gráfico de la pág. 37 están resumidas las condiciones que se han de dar antes de proceder a la instalación de un ordenador.

Clasificación de los ordenadores

Hasta ahora hemos utilizado el término genérico ORDENADOR para referirnos a una gran variedad de máquinas.
Intentemos ahora clasificarlas según una escala de posibilidades y potencialidades.
Las principales características de estas máquinas son:

A / Velocidad de cálculo.
B / Velocidad de transferencia de datos.
C / Cantidad de datos que pueden ser memorizados.
D / Cantidad de usuarios que pueden ser atendidos simultáneamente.

Los valores de estos parámetros definen el campo de aplicación y el tipo de la máquina.
Los principales tipos son:

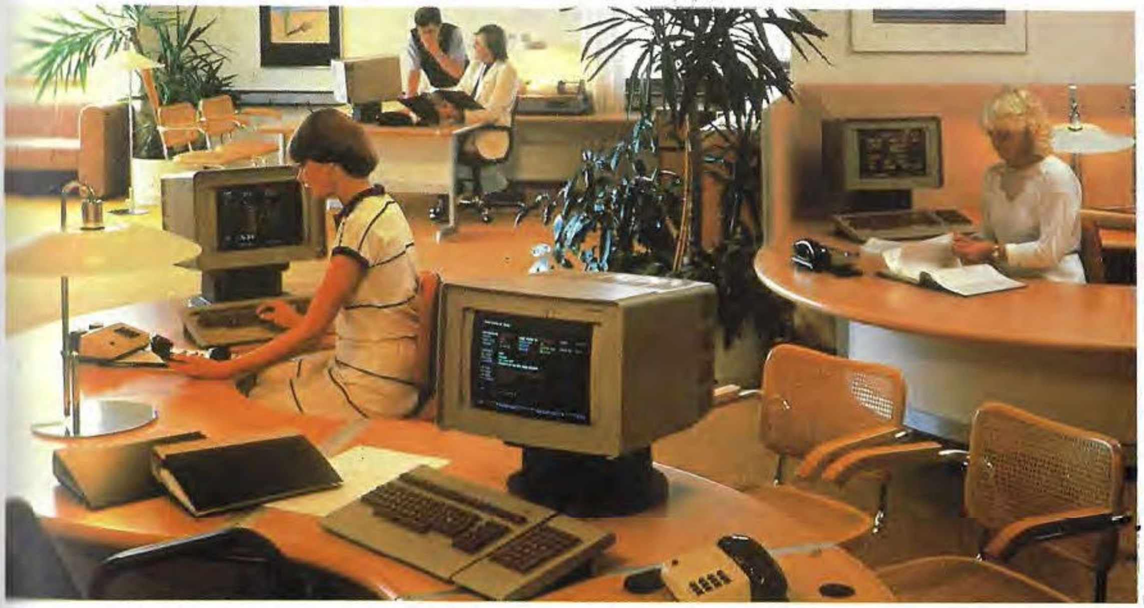
- **MAIN FRAME**
- **MINIORDENADORES**
- **MICROORDENADORES y ORDENADORES PERSONALES**

El MAIN FRAME es un ordenador de grandes dimensiones, que puede servir simultáneamente a decenas de usuarios.
Para su funcionamiento es necesaria una estructura muy compleja. Ha de poseer enormes capacidades de memoria y una gran velocidad.
Estas máquinas se emplean principalmente en la gestión y administración de grandes empresas o para resolver problemas científicos especialmente complejos.
Con los MAIN FRAME se realizan, por ejemplo, los cálculos de las órbitas de los satélites o las trayectorias de los misiles, los cálculos y las comprobaciones de ingeniería para estructuras complejas, o la planificación de circuitos.
En el campo de la gestión, se utilizan para archivos de grandes dimensiones; por ejemplo, en los tribunales, recopilan todas las sentencias pronunciadas. Se emplean también en los registros civiles y tributarios.

Un ordenador personal en el hogar.



Grazia Neri/Contact



Ericsson

Algunas terminales que, en conexión con un ordenador central, permiten el acceso de todos los usuarios al banco de datos de la empresa.

Existen, además, usos muy específicos, como la simulación de guerras. En este caso, grandes ordenadores convenientemente programados pueden simular todas las fases de una hipotética guerra y permiten desarrollar a priori los planes estratégicos, teniendo en cuenta todas las eventualidades, hasta prever su resultado.

El MINIORDENADOR es, por el contrario, una máquina de costo relativamente bajo y más fácil de utilizar, que, normalmente, puede servir a varios usuarios a la vez.

Estas estructuras son sumamente «expandibles», en el sentido de que pueden acoger unidades suplementarias.

La tendencia de los constructores es la de suministrar, inicialmente, unas potencialidades limitadas a las necesidades del momento, ofreciendo seguidamente las unidades suplementarias.

La potencialidad de un sistema en términos de capacidad de memoria y dispositivos de entrada y salida, se denomina CONFIGURACIÓN.

El campo de empleo de estas máquinas es muy variado y depende de su configuración. En los sistemas más desarrollados se acerca a la categoría superior, y en los de base, a la inferior. En general, se utilizan para cubrir la franja de usuarios cuyas necesidades no pueden ser satisfechas por un microordenador, pero que no han de desarrollar un trabajo tan masivo como para precisar un MAIN FRAME.

El MICROORDENADOR (u ORDENADOR PERSONAL) es una máquina empleada por un solo usuario y de utilización muy sencilla.

Como veremos, la frontera entre MICRO y MINIORDENADORES es muy vaga, debido a la aparición de nuevas tecnologías que potencian el sector de los MICROORDENADORES.

A veces se tiende a clasificar los MICROORDENADORES y los ORDENADORES PERSONALES como dos tipos distintos, pero en esencia se trata de máquinas que difieren sólo en la capacidad de memorización de datos.

El empleo de este tipo de máquinas está muy generalizado para todo tipo de aplicaciones menores. Los problemas que pueden resolver son los mismos que los otros tipos; las únicas diferencias son su menor velocidad y la cantidad limitada de datos que pueden memorizar. Como compensación, su bajo costo y sencillez de uso permiten su aplicación a un número de actividades cada vez mayor.

Los ejemplos de actividades automatizadas con estos pequeños ordenadores son innumerables; por ejemplo:

- Distribución de los sueldos en pequeñas empresas.
- Contabilidad comercial.
- Gestión de condominios.
- Cálculos de ingeniería.
- Pequeñas bibliotecas.
- Gestión de agendas.
- Análisis financieros.

El ordenador en la familia

Terminal doméstico, «home computer», microordenador casero: todo ello está configurando una de las aplicaciones más revolucionarias de la cibernética, que penetrará de forma difusa en las familias ya en los años ochenta, estableciendo una comunicación biunívoca con ámbitos marcadamente diversificados y de forma personalizada. El desarrollo de la tecnología y los intereses económicos lo garantizan. Estas son las previsiones en Estados Unidos: se pasará de los 1,5-2 millones de microordenadores hoy en funcionamiento a 5 millones en 1985 y a 10-15 a finales de la década, lo que corresponde a 30-50 millones de consumidores potenciales de estos servicios informativos domésticos; es decir, prácticamente todas las familias estadounidenses dispondrán de ellos.

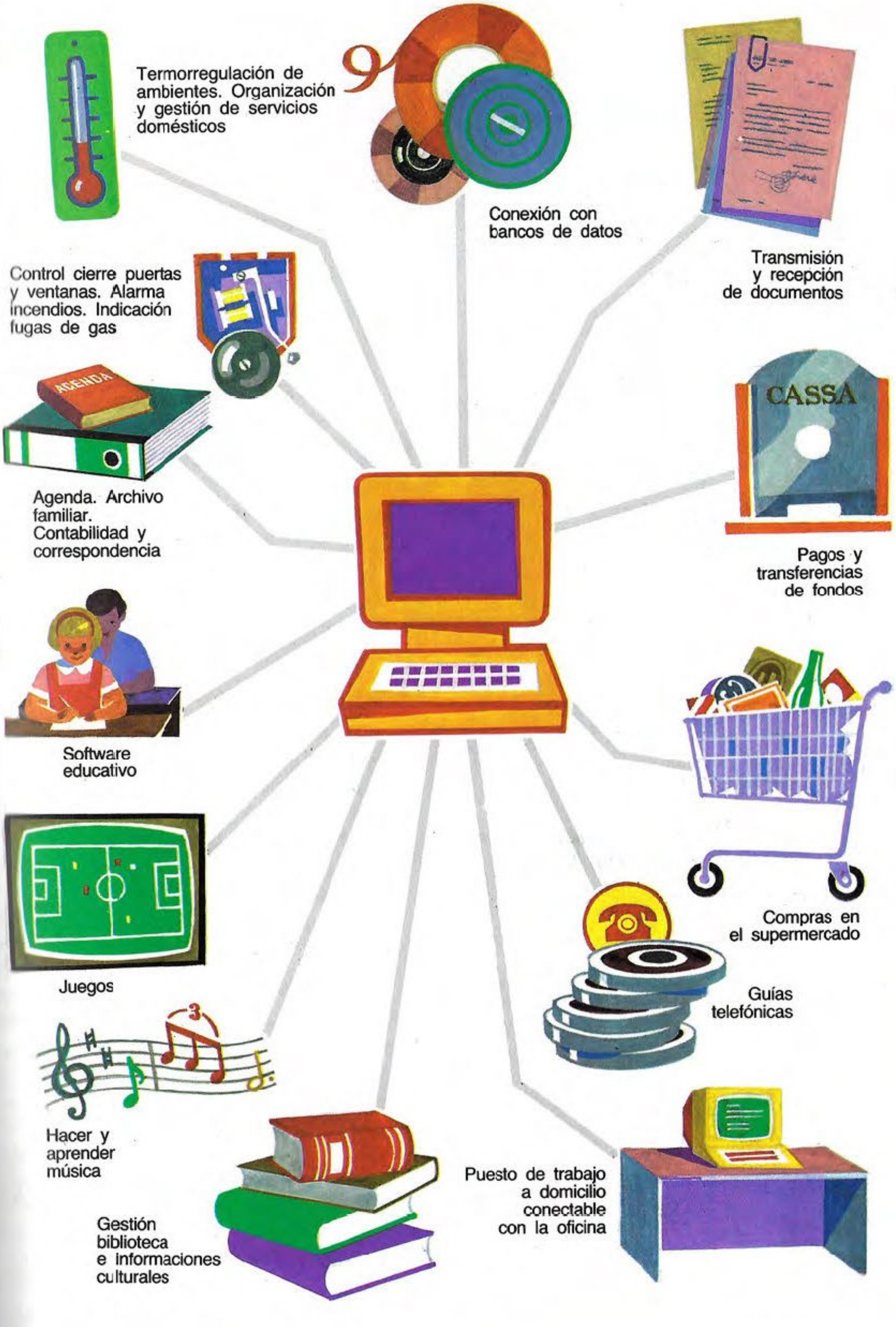
Ya es fácil imaginar las operaciones bancarias efectuadas con un terminal doméstico; al respecto, se podría hablar de la docena de proyectos ya en fase de experimentación en los Estados Unidos, consorciados en ternas de excepcional poder financiero y organizativo: bancos importantes, sociedades telefónicas y televisivas, grupos editoriales de primera fila. Todo esto permite prever que se programará las compras desde casa: técnicamente, se empieza a hablar de shop-at-home; asimismo, los servicios de información prometen ser sumamente diversificados: algunos que ya están funcionando en Estados Unidos suministran informaciones procedentes de más de un millar de bancos de datos, entre los que puede ser muy significativo citar los relativos a informaciones profesionales de carácter médico o agrícola, por dar alguna referencia concreta.

Sin embargo, los sectores de aplicación que más pueden impresionarnos son los realmente biunívocos; por ejemplo, informaciones sobre la seguridad personal, ya sea para emergencias médicas o para protegerse de intrusos o ladrones, o en caso de incendio o inundación. O bien sondeos de opinión de ejecución instantánea, «en tiempo real» cabría decir, que elaboran los resultados mientras tienen lugar las respuestas. Entre las televisiones privadas norteamericanas, han encontrado su espacio propio y funcionan eficazmente las gestionadas por grupos étnicos, pequeñas comunidades o asociaciones religiosas; el terminal doméstico y su posibilidad de comunicación biunívoca se utiliza sobre todo para la recogida inmediata, «en caliente», de

El uso del ordenador personal en casa puede resolver multitud de pequeños problemas: desde la agenda semanal y la lista de las compras a efectuar, hasta la gestión de los sistemas de seguridad.



Dominique Fradin



Termorregulación de ambientes. Organización y gestión de servicios domésticos
Conexión con bancos de datos
Transmisión y recepción de documentos
Control cierre puertas y ventanas. Alarma incendios. Indicación fugas de gas
AGENDA
CASSA
Agenda. Archivo familiar. Contabilidad y correspondencia
Pagos y transferencias de fondos
Software educativo
Compras en el supermercado
Juegos
Guías telefónicas
Hacer y aprender música
Puesto de trabajo a domicilio conectable con la oficina
Gestión biblioteca e informaciones culturales



Dominique Fradin

La aplicación de los ordenadores a programas educativos abre interesantes perspectivas.

suscripciones benéficas. Más corrientes y previsibles, pero sin duda destinadas a obtener gran éxito, resultan otras aplicaciones: el estudio casero, la versión moderna del autodidacta o el ratón de biblioteca, que de esta manera podrá quedarse en su casa; los programas de diversión, por ejemplo los deportivos, que podrían ser seleccionados y programados.

Estos servicios, que pueden considerarse seguros e inscribirse entre los comportamientos del futuro inmediato, complementados por todo lo que la fantasía del hombre y la lógica del lucro sean capaces de imaginar, permiten entrever cuánto más incisiva será esta escalada cibernética con respecto a la aparición de la televisión misma. Ésta implicaba un programa rígido, igual para todos, masificante, concentrado en horarios que a menudo paralizaban la vida social. El microordenador es, por el contrario, un instrumento de elección, de personalización, de libertad y también de democracia.

Por otra parte, podemos intuir las disputas entre todos los miembros de la familia alrededor de ese único instrumento, versátil, sí, pero precisamente por eso totalizador. Sería como concentrar en un único aparato teléfono y enciclopedia, diccionarios y equipos de alta fidelidad, televisor y proyector cinematográfico, agenda y libro de cuentas, lista de la compra y ajedrez... todos disputándose el teclado del microordenador para uso local, para comunicarse o telejugar.

Si por un lado se considera esta posible incidencia en la vida doméstica, no se puede, por otro, dejar de pensar en la respuesta de la tecnología para obviar esta eterna contienda casera. La solución la ofrece el ordenador mismo, que a través de tantas generaciones de sucesivos perfeccionamientos ha trazado el camino del progreso también para los microordenadores: se trata de la multiprogramación, o sea, la posibilidad de desarrollar varios programas simultáneamente; ello presupone más teclados y más pantallas. Entre las paredes mismas del hogar, aparecen pequeñas redes caseras.

Estas reflexiones están basadas en datos, proyectos, tendencias y previsiones ligados a la realidad del mercado y las costumbres estadounidenses. Reconsiderar y replantear estas experiencias desde una perspectiva europea tiene, sin embargo, pleno sentido, pues la informática está evolucionando en todos los países industrializados de manera sustancialmente proporcional a la renta per cápita. A partir de innovaciones surgidas en niveles de cultura técnica y capacidad industrial un tanto distintos, cabe decir que es lógico prever un defasaje temporal de pocos años, que por otra parte se pueden dedicar a la preparación y se recuperarían con la tan desatendida instrucción sobre informática.

(Fragmento de LA INFORMÁTICA A DOMICILIO, de Mauro Langfelder. Feltrinelli, Milán, 1983.)

Los modelos más pequeños encuentran eficaces aplicaciones también en el hogar; pueden servir como agenda, ayudar en los estudios y en el control del gasto familiar. Son, en la práctica, un útil «electrodoméstico» (algunos modelos valen mucho menos que un televisor).

Redes de ordenadores

Todos los ordenadores han de estar en comunicación con el usuario, y por tanto disponen de circuitos capaces de transmitir y recibir datos. En la forma más simple, la transmisión tiene lugar hacia una impresora y un display y desde un teclado; pero nada nos impide conectar, en el lugar de una de estas máquinas, otro ordenador.

Se pueden así montar redes de ordenadores. El objetivo principal de una red consiste en la utilización concentrada de los archivos.

Como se ha visto anteriormente, la utilidad del ordenador se revela sobre todo en la gestión de grandes cantidades de datos (archivos); pero, precisamente por sus dimensiones, los archivos son costosos, debido a su continua necesidad de actualización. Conviene, por tanto, tener un archivo centralizado y un cierto número de usuarios en conexión mediante una red.

La estructura típica de una red se muestra en el gráfico inferior.

En el esquema se evidencian las principales soluciones:

- El satélite* 1 está conectado directamente al ordenador principal; este tipo de conexión se denomina «conexión en estrella».
- Los satélites 2 y 3, también conectados «en estrella», tienen un concentrador.

Este método se utiliza para agilizar el tráfico de datos en largas distancias.

En el ejemplo, los dos satélites (2 y 3) y el concentrador están en una localidad muy distante del ordenador principal.

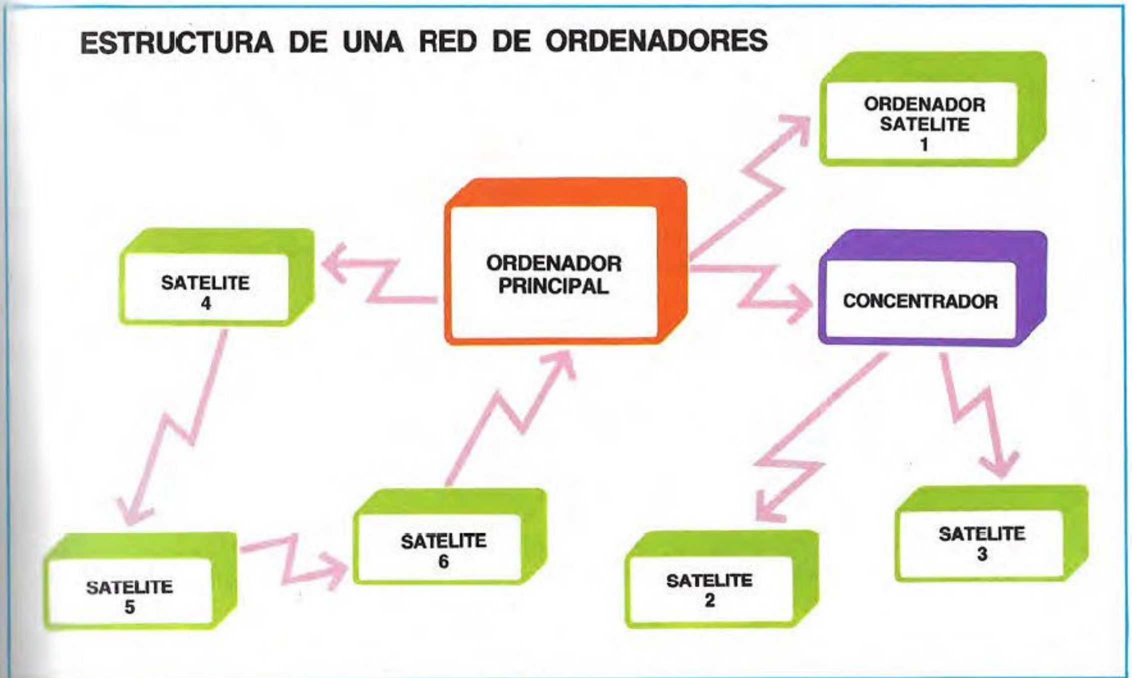
El empleo del concentrador permite reunir un determinado número de líneas locales en una única línea remota.

El concentrador cumple la función de dirigir el intercambio de datos en la línea remota y de distribuirlos entre los satélites. De este modo la línea de conexión de larga distancia es única; luego se subdivide en un determinado número de líneas locales (una por cada satélite).

En el mismo gráfico, los satélites 4, 5 y 6 están conectados «en anillo».

Con este tipo de conexión se puede eliminar el concentrador, pero el sistema deja de funcionar en caso de avería de cualquiera de los satélites o de la línea.

* Con el término «satélite» se designa un ordenador o, en general, un dispositivo cualquiera que realice funciones total o parcialmente dirigidas por otra máquina.





El gráfico de la pág. 43 reproduce el métodc de conexión más común: el de las líneas telefónicas.

Existen dos tipos posibles de transmisión de datos mediante línea telefónica:

- — Por línea conmutada
- — Por línea dedicada

La línea conmutada consiste en la utilización de una línea telefónica normal. La conexión se efectúa igual que una llamada telefónica corriente.

Su ventaja consiste en el bajo costo, mientras que su principal desventaja es la escasa velocidad de transmisión.

La línea dedicada (punto–punto) consiste en un cable que conecta los dos ordenadores.

Los costos son netamente superiores, pero esta desventaja viene compensada por una velocidad de transmisión mucho mayor que, en algunos campos, es esencial.

Estos conceptos pueden resumirse en algunos puntos fundamentales:

Un dispositivo digital puede reconocer sólo dos estados:
ON y OFF
El ordenador digital puede realizar sólo 4 funciones principales:
OPERACIONES ARITMÉTICAS
COMPARACIONES
TRANSFERENCIA DE DATOS
MEMORIZACIÓN DE DATOS
Un programa es una serie lógica de funciones elementales.
El empleo de un ordenador sólo es posible si los términos del problema están exactamente definidos.

Por último, en el gráfico superior se muestra el esquema de un sistema completo.

Las funciones realizadas por cada componente serán analizadas más adelante.

Los sistemas de numeración

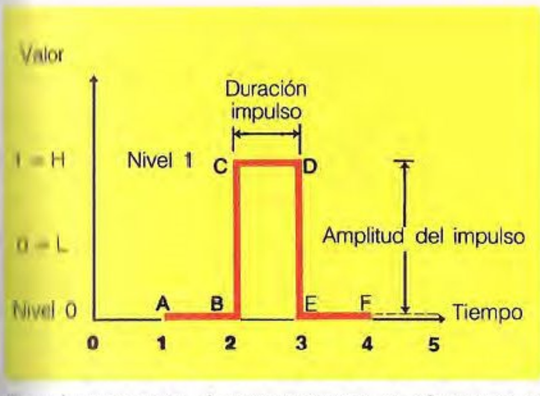
Este capítulo está dedicado a la exposición de los métodos para representar nuestras simbologías (números y letras) en la forma más afín a la estructura del ordenador.
El tema es uno de los más abstrusos desde el punto de vista conceptual, puesto que obliga al lector a abandonar su lógica de matices, reflexiones e intuiciones, para entrar en la rígida lógica de la máquina, que excluye toda forma de sobrentendido.
En la lectura de este tema se recomienda poner la mayor atención en el aspecto conceptual, mientras que el aspecto técnico del desarrollo de un cálculo en la forma propia del ordenador, aunque aparatoso, es básicamente sencillo y podrá ser asimilado posteriormente, a través de los numerosos ejemplos propuestos.

Los impulsos

Un dispositivo digital está capacitado para reconocer sólo estos dos estados:

ON = Presencia de señal
OFF = Ausencia de señal

El punto de referencia es la situación OFF = 0, y se denomina NIVEL 0.
La situación ON = 1 se representa con un nivel de altura proporcional al valor de la señal y se denomina NIVEL 1.
Estas dos situaciones pueden representarse gráficamente de la manera siguiente:

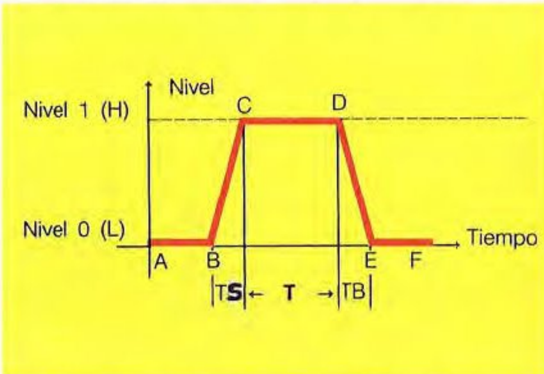


En el momento A empezamos a observar el fenómeno.

En el tiempo A-B (por ejemplo, 1 segundo: del segundo 1 al 2) el nivel se mantiene bajo = 0 (el nivel 0 se puede indicar también con la letra L).
En el tiempo B-C (muy pequeño) se pasa del nivel 0 al nivel 1 (el nivel 1 se indica normalmente con la letra H).
El nivel H se mantiene durante el tiempo C-D.
En el intervalo (muy pequeño) D-E se pasa del nivel H al nivel 0.
El estado de nivel 0 puede durar indefinidamente. El ciclo:

- Nivel 0
- Nivel H
- Nivel 0

se denomina IMPULSO.
En realidad, la forma de un impulso no puede ser tal como se ha descrito.
En la figura anterior, el tiempo necesario para pasar del nivel 0 (L) al nivel 1 (H) no se toma en consideración, y tampoco el tiempo necesario para el paso inverso H-L (puntos D-E).
En realidad, ningún circuito electrónico puede cambiar de estado en un tiempo nulo.
Por lo tanto, la forma real de un impulso es:



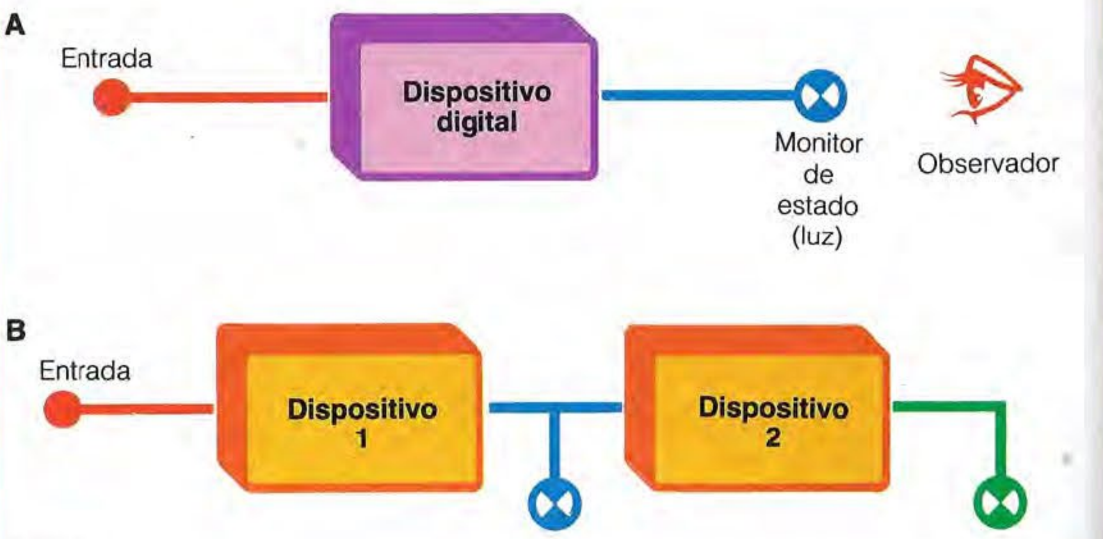
Por comodidad de representación, los tiempos se han dilatado y el punto A se ha hecho coincidir con el tiempo 0.
El tiempo necesario para pasar del nivel 0 al nivel 1 (puntos B-C) se denomina TIEMPO DE SUBIDA (en la figura, TS).
La duración del impulso en el nivel 1 (T) equivale al segmento C-D.
El tiempo necesario para pasar del nivel 1 al nivel 0, entre los puntos D-E, se denomina TIEMPO DE BAJADA (TB).
La línea B-C, es decir, la parte del gráfico correspondiente al tiempo de subida (TS), se denomina FRENTE DE SUBIDA.
Análogamente, el trecho D-E (en el tiempo TB) se denomina FRENTE DE BAJADA.

Resumiendo, los principales parámetros que caracterizan un impulso son:

- **Duración = Tiempo T (trecho C-D)**
- **Tiempo de subida = Tiempo TS (trecho B-C = Frente de subida)**
- **Tiempo de bajada = Tiempo TB (trecho D-E = Frente de bajada)**

Conmutación

Un ordenador sólo puede utilizar, para su aritmética, los dos símbolos ON y OFF.
El estado ON se indica con un 1 (o con la letra H), y el estado OFF con un 0 (o con la letra L).
Supongamos ahora que tenemos un dispositivo digital capaz de mostrarnos su estado (por ejemplo, con un indicador luminoso) y dotado de una entrada, como en el gráfico A; inicialmente, sea 0 (luz apagada) el estado de salida, con la entrada también en 0.
Poniendo la entrada en 1, el dispositivo digital cambiará de estado y la salida se pondrá en 1 (luz encendida).
El observador podrá decir que ha llegado un impulso.
Este cambio de estado se denomina CONMUTACIÓN.
El dispositivo ha conmutado una vez.
Poniendo de nuevo a 0 y luego a 1 la entrada, el dispositivo tendrá una nueva conmutación y la salida pasará de 1 a 0.
Puesto que el observador recuerda haber visto antes un estado 1, puede decir que el actual estado 0 es consecuencia de una nueva conmutación. El dispositivo, por tanto, ha conmutado dos veces, es decir, ha contado 2.
Supongamos ahora que el observador inicie su experimento en el momento del segundo impulso de entrada.
En este caso verá la salida cambiar de 1 a 0 y no podrá determinar si ha habido una sola conmutación o dos. Esta incertidumbre desaparece si a la salida del dispositivo, además del indicador luminoso, ponemos un segundo dispositivo idéntico, como se ve en el gráfico B.
La salida del dispositivo 1 provoca la conmutación del dispositivo 2 y, por lo tanto, se puede determinar el número de impulsos de entrada (0, 1, 2), ya que el segundo dispositivo memoriza y evidencia los cambios de estado del primero.



Número impulsos en Entrada	Estado 1.ª salida	Estado 2.ª salida
0	OFF	OFF
1	ON	OFF
2	OFF	ON

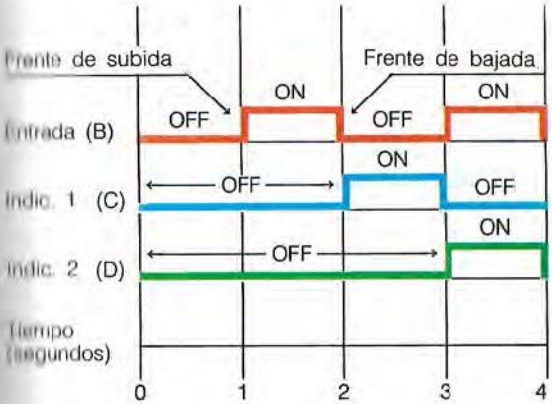
Los estados de Entrada, Indicador 1 e Indicador 2 pueden ordenarse en la tabla siguiente:

-A- Impulso Número	-B- Entrada	-C- Indicador 1	-D- Indicador 2
0	0	0	0
1	1	1	0
2	1	0	1

Esta tabla muestra el funcionamiento del sistema desde el punto de vista lógico y se llama

TABLA DE LA VERDAD

Cada circuito digital tiene su propia tabla de la verdad, más o menos complicada según las funciones que el circuito ha de realizar.
Supongamos que, en la entrada, los impulsos lleguen al ritmo de uno por segundo.
Podremos construir el diagrama siguiente:



- El indicador 2 (D) conmuta al segundo impulso de entrada (B); tenemos, por tanto, un contador para 2.
- La entrada (B), constituida por una serie de impulsos equidistantes en el tiempo, se denomina CLOCK.

Obsérvese que:

EL CAMBIO DE ESTADO SE PRODUCE CUANDO EN LA ENTRADA SE PASA DE ON A OFF.

Las bases de numeración

Existen distintos sistemas de numeración; el más afín a nuestra lógica es el sistema posicional.
En este sistema cada número tiene un «peso», es decir, un valor, que depende de su posición. Si escribimos el número 5 queremos decir 5 unidades, pero cambiando la posición del símbolo 5 podemos indicar 5 decenas (50), 5 centenas (500), etc. El símbolo 5 adquiere un valor diferente según su posición; cada posición tiene un valor correspondiente al valor intrínseco del número (0, 1, 2, 3, 4, 5, 6, 7, 8 o 9) multiplicado por 10 tantas veces como indica la posición, de acuerdo con la tabla siguiente:

	Centenas	Decenas	Unidades
Peso	100	10	1
Exponente	5×10^2	5×10^1	5×10^0
Valor posic.	5×100	5×10	5×1
Valor	500	50	5

A cada cambio de posición, el número se multiplica por 10. El valor 10 es la «base» de nuestro sistema de numeración, que por esta razón se denomina **sistema posicional en base diez** (numeración árabe).
Un ejemplo de sistema no posicional lo constituye la numeración romana.
En ella, al valor 5 le corresponde el símbolo V, mientras que al valor 50 le corresponde el símbolo L. Para pasar de 5 a 50 no basta con cambiar la posición del símbolo 5 (V), sino que hay que introducir uno nuevo (L).
Este tipo de numeración, incómodo de leer y poco adecuado para la realización de cálculos, fue abandonado y en su lugar se adoptó universalmente el sistema posicional.
El sistema posicional más común es el decimal: en él los símbolos son diez y la base es diez.
En los ordenadores no es posible utilizar diez símbolos distintos. Como ya hemos visto, la máquina sólo puede reconocer dos estados: el estado ON (nivel alto) y el estado OFF (nivel bajo); sólo dispone, pues, de dos símbolos: **1** (ON) y **O** (OFF).
Por ese motivo hay que utilizar un sistema de numeración que sólo requiera el uso de los símbolos 1 y 0; en función de esta característica, el sistema se denomina **sistema binario.**
Este tipo de numeración, en contra de lo que pueda parecer, no ha surgido con los ordenadores, sino que ya se utilizaba en China hacia el siglo XII. El sistema binario sigue siendo un sistema posicional, puesto que el valor de los símbolos depende de su posición.

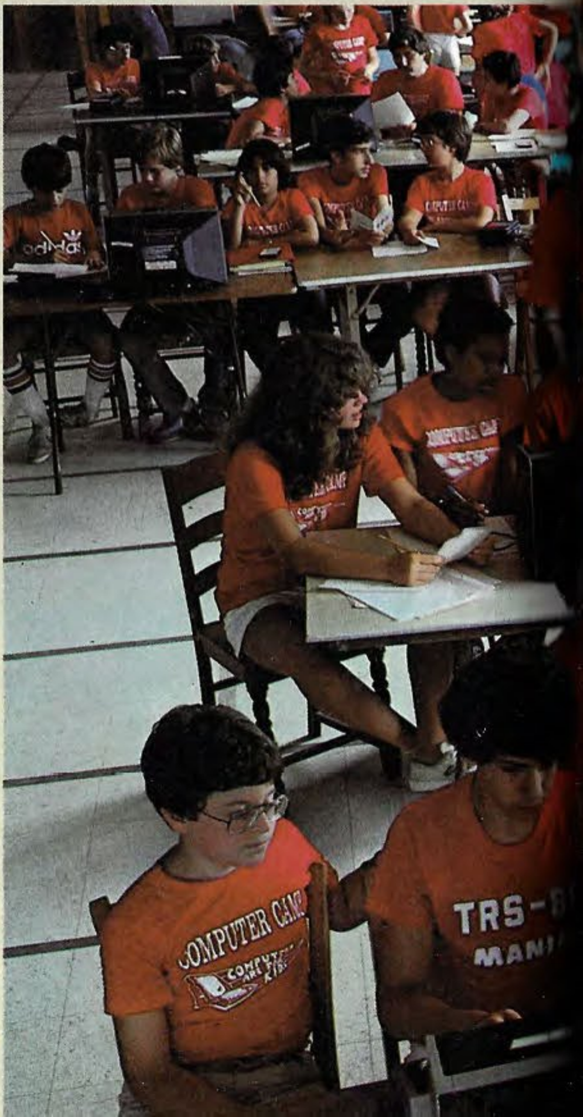
El ordenador en el colegio

El ordenador puede ser una máquina extraordinaria para enseñar; pero, ¿para enseñar qué? Cualquier cosa: no sólo informática y todas las demás materias científicas, como matemáticas, química o biología, sino también las que tradicionalmente no pertenecen al grupo de las ciencias, como idiomas o música. Los niños, por otra parte, parecen tener una predisposición natural para familiarizarse con el uso de un ordenador personal. Tomemos el ejemplo de un programa muy sencillo, para niños de la escuela elemental: el programa de la enseñanza de la resta; en el diálogo siguiente, los alumnos obtienen una reacción inmediata a sus respuestas:

Ordenador: 3–2 = ?
Niño: 1
Ordenador: Muy bien. 7–2 = ?
Niño: 5
Ordenador: Estupendo. ¿Y 23–16?
Niño: 6
Ordenador: ¡Oh, oh! ¿Quieres intentarlo otra vez
Niño: 7
Ordenador: Ahora está bien. ¿Y 44–38?

A cada respuesta correcta, el alumno es elogiado, y si se equivoca se le invita a repetir.
Las frases «Muy bien», «Estupendo», «¡Oh, oh!» etc., utilizadas en el ejemplo de la resta, no son más que propuestas esquemáticas.
Reacciones más pintorescas, por ejemplo un rostro sonriente o afligido, o incluso dibujos animados con efectos sonoros (como un toque de trompeta para las respuestas correctas y un chirrido de frenos o el ruido de un choque brusco para los errores), son más eficaces y divertidas, siempre que las reacciones ante una respuesta equivocada no sean demasiado emocionantes y perturben la atención.
El tipo de presentación elegido por el programa puede variar enormemente. En el sencillo ejemplo anterior, el programa presenta problemas cada vez más difíciles, tras comprobar que el alumno es capaz de resolver los más fáciles. Por eso el programa deja atrás rápidamente las restas entre números de una sola cifra, como 3–2, 7–2, que son resueltos fácilmente. Esta decisión puede derivar de factores menos evidentes que el mero hecho de que el alumno haya dado dos respuestas correctas seguidas; el reloj incorporado al ordenador puede, por ejemplo, tener en cuenta también el tiempo empleado para contestar las dos preguntas; por otra parte, los datos de la lección anterior pueden ser conservados para recordarle al alumno (si lo hubiera olvidado) que ya había resuelto antes problemas de este tipo.
En cualquier caso, el problema con dos cifras (23–16), más difícil, es propuesto luego, seguido de «44–38», y así sucesivamente. Además de proponer problemas de dificultad creciente, el programa puede ofrecer al alumno la posibilidad de cambiar de tema (pasando tal vez de las restas a un programa de lecturas poéticas, de composición de música rock de partido de fútbol simulado) o de poner fin a la lección, especialmente si el tiempo entre una respuesta y otra o la calidad de las mismas indica que el estudiante empieza a estar cansado o aburrido.
Aunque elemental, el ejemplo anterior utiliza, para restar, un método cualitativamente muy superior al de los libros de texto clásicos.

Grazia Neri/E. Ferorelli



Efectivamente, cuando el estudiante está usando un libro de texto, sólo puede resolver los problemas en el orden en que están escritos, sean o no adecuados a su nivel (si son demasiado fáciles o demasiado difíciles, aprende muy poco). Además, el estudiante no recibe ninguna información sobre la exactitud de sus respuestas hasta que no vuelve la página, o ni siquiera entonces. Si el ejercicio corregido, con indicaciones sobre las respuestas correctas y las equivocadas, le es devuelto días después, es probable que ni se acuerde de qué se trataba. Tanto el interés como el nivel de aprendizaje se mantienen mucho más altos si el estudiante obtiene la respuesta correcta a medida que resuelve los problemas. Hay que tener en cuenta, por otra partc, quc los métodos tradicionalcs no sc limitan a la escuela elemental y las restas. Basta entrar en una clase de enseñanza media para darse cuenta de que el profesor suele disertar sobre un tema preestablecido sin tener demasiado en cuenta las preguntas de los alumnos. Éstos copian sus apuntes palabra por palabra, incluidas, probablemente, las tareas para hacer en casa, que requerirán varios días de trabajo y que repasarán, tal vez, sólo semanas después. El alumno que haya cometido algún error, en el momento en que se le dé la solución exacta podría no tener ya idea de por qué su respuesta no era correcta. No hay motivo para que las mismas técnicas de reacción sencillas que facilitan la enseñanza de la aritmética a los niños, no sirvan también para aumentar la eficacia de lecciones sobre otros temas, desde el cálculo integral o los problemas geométricos a la lengua de Shakespeare. De hecho, un nuevo método pedagógico está surgiendo, gracias al empleo de los ordenadores personales, en las escuelas de Estados Unidos y en las de algunos países europeos.

(de THE ELECTRONIC COTTAGE, © Joseph Deken, 1981)

Hora de clase, con ayuda de ordenadores, en una escuela estadounidense.



Examinemos ahora con detalle los mecanismos del sistema posicional. Como hemos visto, en la notación decimal que usamos habitualmente, todo número se representa mediante una sucesión ordenada de las cifras 0, 1, 2, 3, ..., 9. El valor de una misma cifra, por ejemplo de la cifra 5, es distinto según la posición que ocupe en el número; así, 50 no es lo mismo que 500.
Cuando escribimos un número en base 10 realizamos una abstracción dictada por la costumbre, pero en realidad utilizamos las cifras de que disponemos como multiplicadores de las potencias sucesivas de la base 10.
Las potencias de una base son los valores que se obtienen multiplicando entre sí tantos factores iguales a la base como indica el exponente puesto encima y a la derecha. Por ejemplo:

$10^0 = 1$*	potencia cero (unidades)
$10^1 = 10$	potencia uno (decenas)
$10^2 = 10 \times 10 = 100$	potencia dos (centenas)
$10^3 = 10 \times 10 \times 10 = 1.000$	potencia tres (millares)

En un número decimal cada cifra, empezando por la derecha y siguiendo hacia la izquierda, es un multiplicador de las potencias sucesivas de la base 10 (normalmente se habla de unidades, decenas, centenas, millares, etc., pero estos términos dejan de tener sentido cuando la base es distinta de 10).
El procedimiento riguroso que permite interpretar un número decimal y determinar su valor se describe en la tabla siguiente, referida al número decimal 1.024:

Columna	d	c	b	a
Número (cifras)	1	0	2	4
Potencias de la base	10^3	10^2	10^1	10^0
Peso de las posiciones	1000	100	10	1
Valor de las cifras (cifra × peso)	1000	0	20	4
Valor del número	1000+0+20+4=1024			

Normalmente utilizamos la notación decimal, pero esta circunstancia es puramente arbitraria, y se atribuye al hecho de que nuestra capacidad inmediata de visualización de una cantidad se limita al valor diez, es decir, al número de dedos de que disponemos. Una cantidad determinada puede expresarse en cualquier otra notación, escogiendo un valor cualquiera de la base y repitiendo el procedimiento de construcción del número. La elección del valor de la base determina el número de símbolos a utilizar.

BASE 10, los símbolos son diez:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

BASE 2, los símbolos son dos:
0, 1 (o bien L, H)

BASE 8, los símbolos son ocho:
0, 1, 2, 3, 4, 5, 6, 7

BASE 16, los símbolos son dieciséis:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

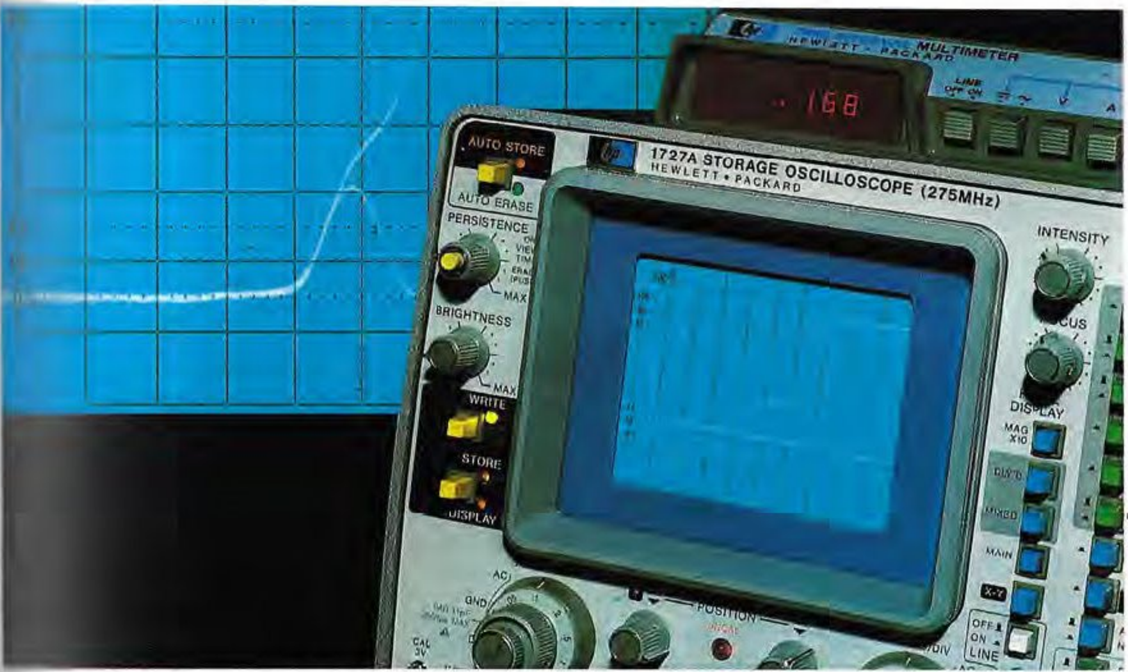
De la base 8 y la base 16 hablaremos detalladamente más adelante; veamos ahora el sistema de numeración en base 2.

Notación binaria

La notación binaria, como la decimal, es posicional: si para escribir e interpretar una cifra decimal realizamos de forma inconsciente las operaciones necesarias, en el sistema binario necesitamos desarrollar todos los pasos, pero las reglas son las mismas.
Profundizaremos en el tema mucho más de lo estrictamente necesario para la programación y, por tanto, lo que sigue puede leerse meramente a título informativo; la práctica necesaria se adquirirá a lo largo de la obra gracias a los numerosos ejemplos expuestos.
Como ya hemos visto, la notación binaria (o sea en base 2) permite expresar cualquier cantidad utilizando sólo los símbolos 0 y 1. En este caso las cifras que ocupan las distintas posiciones se multiplicarán por las potencias sucesivas de la base 2. El procedimiento de interpretación de un número binario (en este caso concreto, el número 1011) se describe en la tabla siguiente, análoga a la utilizada para la base 10:

Columna	d	c	b	a
Número (cifras)	1	0	1	1
Potencias de la base	2^3	2^2	2^1	2^0
Peso de las posiciones	8	4	2	1
Valor de las cifras (cifra × peso)	8	0	2	1
Valor del número	8+0+2+1=11 **(decimal)**			

* Cualquier número elevado a la potencia 0 da 1. Por lo tanto, $2^0 = 1$.



Osciloscopio Hewlett Packard que muestra las formas de onda de señales digitales.

Consideremos otro ejemplo, interpretando el número binario de 8 cifras 11111111:

Columna	h	g	f	e	d	c	b	a
Número (cifras)	1	1	1	1	1	1	1	1
Potencias de la base	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Peso de las posiciones	128	64	32	16	8	4	2	1
Valor de las cifras (cifra × peso)	128	64	32	16	8	4	2	1

Valor del número 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255 **(decimal)**

Como veremos, un conjunto de 8 cifras binarias (cuyo máximo valor decimal es 255) constituye un dato básico en los ordenadores. En ellos, todo número se traduce en una serie de símbolos 1 y 0; sin embargo, el sistema binario requiere una gran cantidad de símbolos para indicar un valor numérico, y resulta, por tanto, incómodo para quien utiliza la máquina. Por este motivo se han introducido otros dos sistemas posicionales: el **octal** y el **hexadecimal.**

Como veremos, estos sistemas permiten conservar una simbología adecuada al funcionamiento del ordenador, pero de forma compacta y, por tanto, de utilización más cómoda que la binaria; permiten, en efecto, escribir los números evitando las larguísimas series de 1 y 0.

La relación entre el sistema binario y los sistemas octal y hexadecimal es la misma que hay entre el tiempo expresado en minutos y el expresado en horas.

Así pues, las 8 de la mañana equivalen a $8 \times 60 = 480$ minutos; por tanto, decir que son las 8 horas o los 480 minutos es lo mismo; la diferencia estriba en que la primera forma es más cómoda.

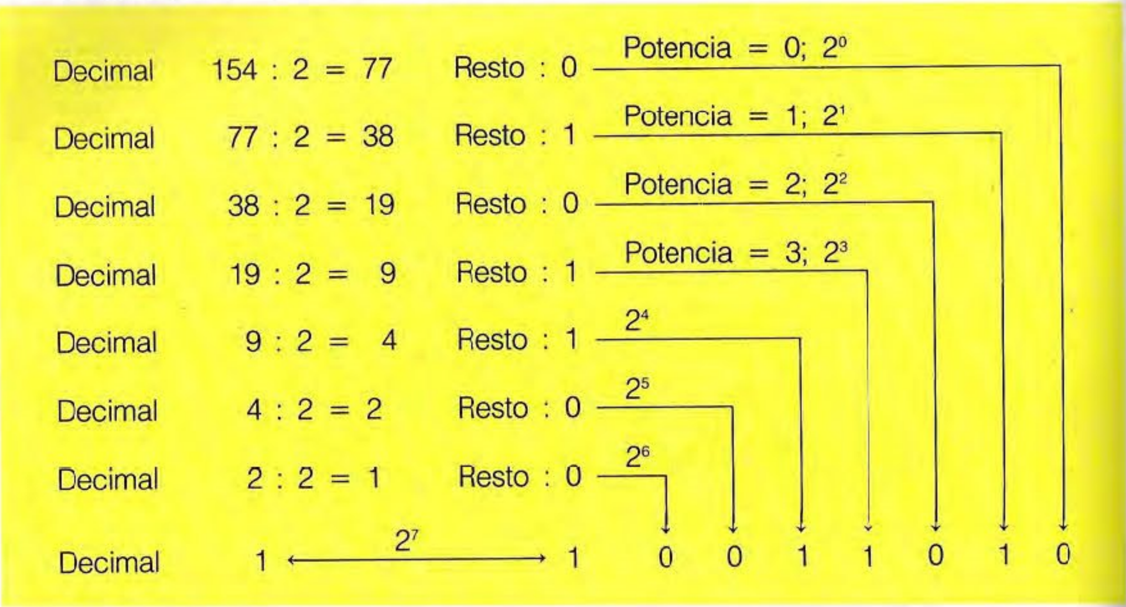
Del mismo modo, el valor de un número es el mismo en notación binaria, octal o hexadecimal; sólo varía la forma en que está expresado.

De todos modos, **el ordenador utiliza siempre y solamente el sistema binario;** por tanto, recordemos que:

UN DISPOSITIVO DIGITAL REALIZA LOS CÁLCULOS EN BASE 2.

Hemos visto cómo se calcula el valor decimal de un número binario. Veamos ahora cómo se obtiene un número binario a partir del decimal. El procedimiento es el contrario del precedente. Se va dividiendo el número decimal por 2 hasta obtener un 1 como cociente. En cada división se escribe 0 si no hay resto y 1 si lo hay.

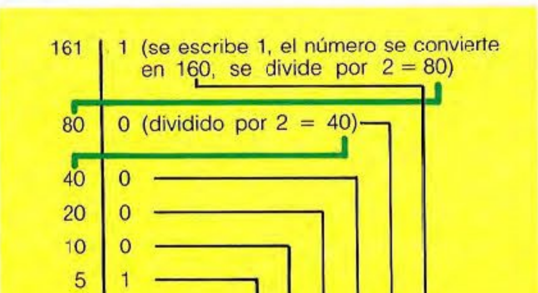
Consideremos, por ejemplo, el número decimal 154:



Escribiendo los restos obtenemos: 1 0 0 1 1 0 1 0, que es el equivalente binario del 154 decimal.
El procedimiento puede ser agilizado de la forma siguiente:

A / Si el número es par se escribe un 0.
B / Si el número es impar se escribe un 1 y se resta 1 del número.
C / Se divide por 2.
D / Si el cociente no es 1, se repite con él el paso **A.**

Por ejemplo, si tomamos el número decimal 161, tenemos:



Para comprobar el resultado, convirtamos el número binario 1 0 1 0 0 0 0 1 en decimal:

1	0	1	0	0	0	0	1
2^7	+	2^5		+			2^0
↓		↓					↓
128	+	32		+			1

Efectivamente, la suma da 161.

Notación hexadecimal

Como hemos dicho, en los ordenadores se usan habitualmente números binarios de 8 cifras o múltiplos de 8 (16, 32). Para indicar tales números hacen falta 8 símbolos (o, respectivamente, 16, 32) que pueden ser, cada uno, un 0 o un 1; esto supone una notación demasiado larga.
Para mayor comodidad se ha introducido una forma de numeración más compacta: **la numeración hexadecimal**, es decir, en base 16. Esto no altera lo dicho hasta ahora: dentro de la máquina, la simbología sigue siendo binaria. **El sistema hexadecimal no es más que un sistema de representación más cómodo.**
Con este sistema de numeración las unidades pueden valer de 0 a 15 (del mismo modo que las unidades decimales valen de 0 a 9); el grupo inmediatamente superior empieza a partir de 16, es decir, del valor de la base (igual que las decenas del sistema decimal empiezan a partir de 10).
Consideremos, por ejemplo, el número binario 1 0 1 0 0 0 0 1 (que, como hemos visto, corresponde al 161 decimal); sabemos que cada símbolo (0, 1) representa una potencia de 2.

Reescribamos el número binario en una tabla que dé las correspondientes potencias de 2:

■ Exponente	7	6	5	4	3	2	1	0
■ Valor de la potencia	128	64	32	16	8	4	2	1
■ Número binario	1	0	1	0	0	0	0	1
		X				Y		

Veamos la mitad de la derecha del número binario 1 0 1 0 0 0 0 1, es decir: 0 0 0 1 (los exponentes son 3, 2, 1, 0 y sus valores 8, 4, 2, 1). Con un tal número binario, la máxima cantidad que se puede escribir (convirtiendo en 1 todas las cifras que lo forman) es:

■ 1 1 1 1 binario = 8+4+2+1 = 15 decimal.

Para escribir el número decimal 16 hay que pasar al grupo siguiente. Resumiendo, pues:

Con un número binario de cuatro cifras se puede escribir, como máximo, el valor 15. Para escribir el valor 16 hay que pasar al grupo de cuatro cifras siguiente (grupo X en la tabla).
Estas consideraciones nos llevan a la conclusión de que agrupando los números binarios de cuatro en cuatro se tiene un sistema de numeración en base 16. Dicho sistema se llama HEXADECIMAL.

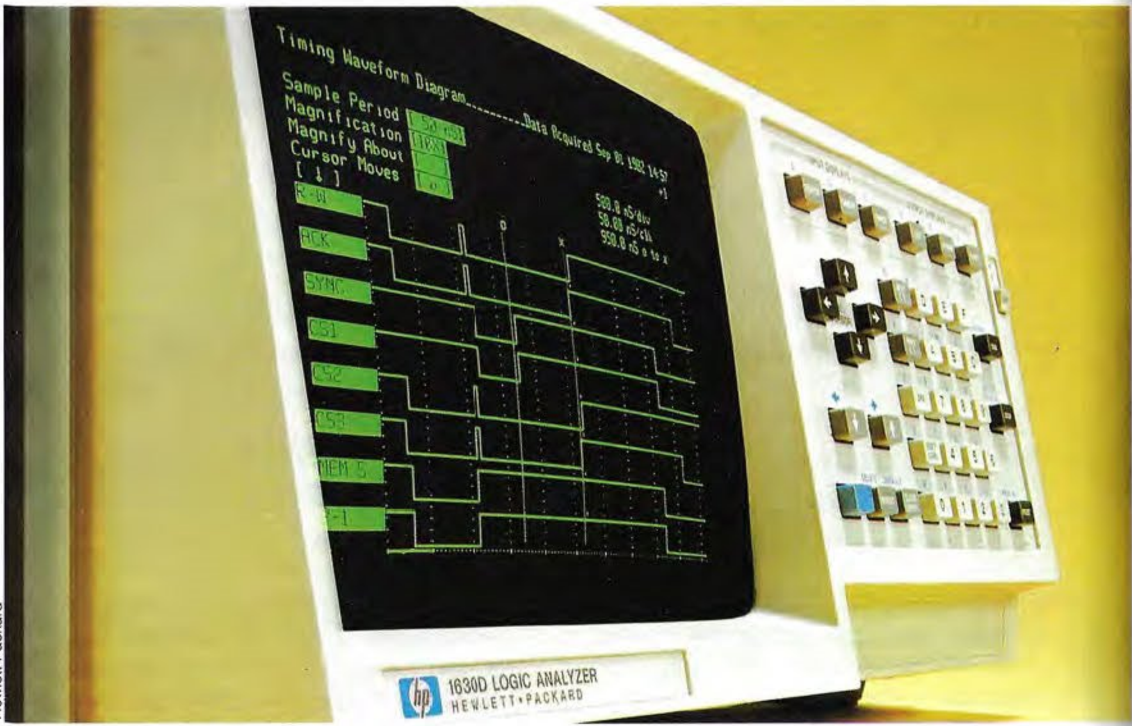
Efectivamente:

- **El grupo de las unidades hexadecimales (grupo Y) permite escribir hasta 15 (como las unidades decimales permiten escribir hasta 9; o sea, una unidad menos que la base, 10).**
- **El grupo inmediatamente superior (grupo X) comienza con el valor de la base, 16.**

Efectivamente, el valor mínimo (aparte de 0) que se puede escribir en el grupo X es 0 0 0 1, que vale precisamente 16. Igual que el valor mínimo del grupo de las decenas es 10.

Adoptando la notación hexadecimal, cada grupo de cuatro cifras binarias tiene valores comprendidos entre 0 y 15.
Es decir, los exponentes de 2 son 0, 1, 2, 3, y las potencias son 1, 2, 4, 8, que sumadas dan 15.
Reescribamos ahora el anterior número binario 1 0 1 0 0 0 0 1 de esta nueva forma:

	Segunda cifra hexadecimal				Primera cifra hexadecimal			
Exponente	3	2	1	0	3	2	1	0
Valor de la potencia	8	4	2	1	8	4	2	1
Número binario	1	0	1	0	0	0	0	1
	X				Y			



Hewlett-Packard

El analizador de estado lógico HP 1630 D visualiza los niveles lógicos de los circuitos digitales.

El cálculo de Y y X en hexadecimal es inmediato:

Y = 1, puesto que sólo está la potencia $2^0 = 1$.
X = 10, puesto que están las potencias 2^3 y 2^1: $8+2=10$.

Por lo tanto, la cifra total se debería escribir:

	X	Y
X Y	**1 0**	**1**

Esta simbología no es clara en tanto que, a priori, no se puede saber si el 10 pertenece todo él al grupo **X** o bien si el grupo **X** vale 0 y existe otro grupo **Z** que vale 1.
Para obviar este inconveniente se ha acordado representar mediante letras las cifras hexadecimales de valor superior a 9, según la tabla:

hexadecimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
símbolo	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

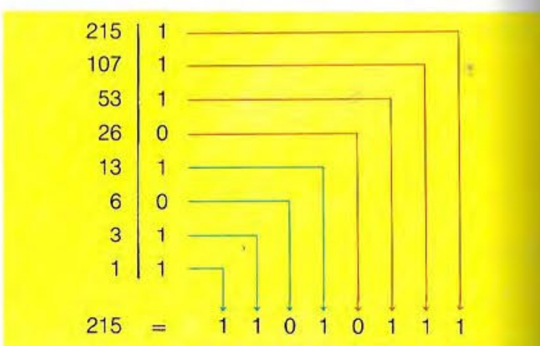
Por tanto, el valor 10 se indica con el símbolo A.
El número binario 1 0 1 0 0 0 0 1 se convierte, pues, en:

1010	0001
A	**1**

Resumiendo:

Valor decimal: 161
Correspondiente binario: 10100001
Correspondiente hexadecimal: A1

Como puede verse, la notación hexadecimal es mucho más compacta.
Para poner otro ejemplo, tomemos el número decimal 215. Convirtámoslo en binario:



Realicemos ahora una tabla de conversión a hexadecimal.
Recordando que una cifra hexadecimal agrupa cuatro cifras binarias, para poder escribir el número 1 1 0 1 0 1 1 1 necesitaremos dos posiciones hexadecimales:

	Segundo hexadecimal				Primer hexadecimal			
Valor de la potencia	8	4	2	1	8	4	2	1
Número binario	1	1	0	1	0	1	1	1
	8+4+1 = 13, es decir = **D**				4+2+1 = 7, es decir = **7**			

Tenemos, por tanto:

- decimal : 215
- hexadecimal: **D7**
- binario : 1 1 0 1 0 1 1 1

Como comprobación, procedamos al revés, es decir, del valor hexadecimal **D7** obtengamos el correspondiente valor decimal:

- El valor **7** representa las unidades, y por tanto, no varía.
- El valor **D** = 13 al ser el primer múltiplo de la base, se multiplica por la base 16.

Tenemos, pues, $(13 \times 16) + 7 = 215$ decimal.

Notación octal

Es una notación muy similar a la hexadecimal. Consiste en dividir el número binario en grupos de tres cifras, cuyos valores son 4, 2, 1 (exponentes 2, 1, 0). Operando en esta nueva base se obtiene, con el número binario 1 0 1 0 0 0 0 1 = 161 decimal, la tabla:

	Tercera cifra octal			Segunda cifra octal			Primera cifra octal		
Exponente	2	1	0	2	1	0	2	1	0
Valor de la potencia	4	2	1	4	2	1	4	2	1
Número binario		1	0	1	0	0	0	0	1
Valor octal	**2**			**4**			**1**		

Se tienen las correspondencias:
decimal : 161
binario : 1 0 1 0 0 0 0 1
hexadecimal: **A1**
octal : **241**

Análogamente a lo que hemos visto en la notación hexadecimal, para convertir el número octal **241** en decimal se siguen las reglas:
La unidad permanece invariable.
El número siguiente debe multiplicarse por 8^1.
El siguiente, por 8^2, y así sucesivamente.
Tenemos, por tanto:

Octal	Transformación	Decimal
1	–	**1**
4	**x 8**	**32**
2	**x 64**	**128**
	Resultado	**161**

Resumiendo:

- **Los circuitos digitales funcionan con aritmética en base 2.**
- **Un número binario se representa mediante una serie de símbolos 1 y 0.**
- **Cada símbolo representa el multiplicador de la correspondiente potencia de 2.**
- **El valor decimal de una cifra hexadecimal se obtiene multiplicándola por la correspondiente potencia de 16.**
- **En notación octal la base es 8.**
- **El valor decimal de una cifra octal se obtiene multiplicándola por la correspondiente potencia de 8.**

Operaciones en notación binaria

Hemos visto que se han introducido las notaciones octal (base 8) y hexadecimal (base 16) con objeto de hacer más compacta la forma de presentación de los números binarios.

El ordenador, sin embargo, no utiliza directamente estas notaciones: todos los cálculos se realizan en notación binaria.

A menudo, en la escritura de programas hay que controlar a mano los resultados de determinadas instrucciones dadas a la máquina y, por tanto, es necesario conocer el método para efectuar sumas y restas en base 2.

Como veremos, el objeto de estos controles no es el de comprobar la exactitud de los cálculos matemáticos (la máquina realiza los cálculos con mucha más velocidad y exactitud que nosotros), sino el de obtener ciertos resultados en operaciones «lógicas». El significado y la función de las operaciones «lógicas» se verán más adelante. Por ahora intentaremos adelantar estos conceptos mediante un ejemplo.

El ordenador, para poder dialogar con nosotros, ha de contar con una tabla de traducción entre nuestro alfabeto y los símbolos que él utiliza en lugar de nuestras letras y números.

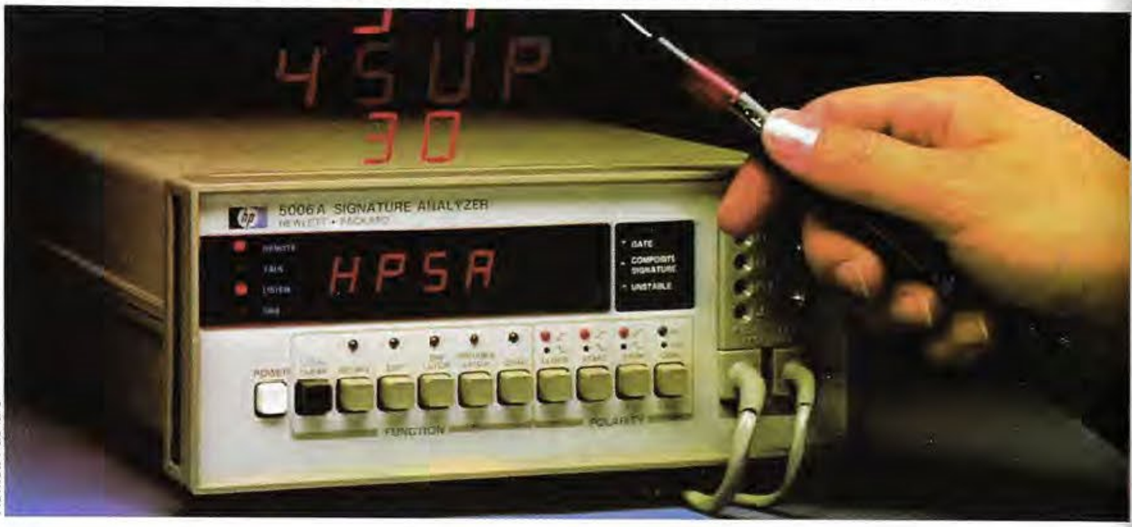
De lo que hemos visto hasta ahora se deduce inmediatamente que un número cualquiera viene representado, en el interior del ordenador, por una serie de símbolos 1 y 0.

Puesto que la máquina no conoce otros símbolos, también las letras del alfabeto han de ser expresadas de esta manera.

En otras palabras, todos nuestros símbolos gráficos son representados, en el ordenador, en forma de código binario. Así, si el usuario quiere introducir en la memoria de la máquina la letra «A», dicha letra será convertida en un número binario. Luego, para obtener el código de la letra «B» basta con sumar 1 al código de la letra «A», y así sucesivamente para todo el alfabeto.

Como veremos, este método permite establecer todos los códigos correspondientes a nuestros símbolos gráficos.

Analizador de firma HP 5006 A: indica si el circuito examinado funciona normalmente.



Hewlett-Packard

TEST 1

Después de cada tema se planteará un test de comprobación.
El lector podrá controlar sus respuestas comparándolas con las soluciones que se dan más adelante.
En general, si al menos la mitad de las respuestas son correctas, se puede proseguir la lectura sin problemas.
Si el resultado es inferior, se aconseja releer el capítulo correspondiente, a no ser que se trate de un asunto marginal y no indispensable. Éste último caso se indica en las soluciones.

1 / ¿Cuáles de estas afirmaciones son ciertas?
- □ a - En los ordenadores se usa la notación binaria porque permite una interpretación fácil por parte del operador.
- □ b - La notación octal se ha introducido para representar los números binarios de forma más compacta.
- □ c - Una cifra hexadecimal puede expresar, como máximo, el valor de 15.
- □ d - En la notación octal no existe la cifra 8 (¿Cómo se escribe en octal el decimal 8?)

2 / Convertir los siguientes números binarios en su correspondiente valor decimal:
1 0 0 1 - 1 1 - 1 - 1 0 1 0 1 - 1 1 0 0 1 - 1 1 1

3 / Convertir los siguientes números binarios en su correspondiente valor octal:
1 1 0 1 1 1 - 1 1 - 1 0 1 0 1 0 - 1 1 1 1 1 1

4 / Convertir los siguientes números binarios en su correspondiente valor hexadecimal:
1 1 0 1 1 1 1 1 - 1 1 0 1 1 0 1 1 - 1 1 1 1 1 1 1 1 - 1 1 1 1 1 1

5 / Traducir al sistema binario los siguientes números decimales:
10 - 100 - 1000

6 / Los cuatro números binarios:
a - 0 0 0 1 - b - 0 0 1 0 - c - 0 1 0 0 - d - 1 0 0 0
se obtienen desplazando el símbolo 1 un lugar a la izquierda cada vez.
- □ ¿Cuál es el valor decimal de cada uno de ellos?
- □ Observando los valores decimales, ¿a qué operación aritmética equivale el desplazamiento a la izquierda del símbolo 1?
- □ Desplazando el símbolo 1 hacia la derecha, ¿qué operación aritmética se efectúa?

7 / ¿Cuáles de las siguientes afirmaciones son ciertas?
- □ a - El frente de subida de un impulso corresponde a la zona en la que la señal pasa de un nivel bajo a uno alto.
- □ b - En la notación hexadecimal, las cifras superiores a 9 se indican con letras del alfabeto.
- □ c - En notación hexadecimal no se puede indicar cifras superiores a 9.
- □ d - Un circuito digital no puede reconocer más estados que la presencia o ausencia de señal.

8 / Hallar el valor decimal de los siguientes números octales:
7 - 70 - 700 - 111

9 / Hallar el valor decimal de los siguientes números hexadecimales:
7 - 70 - A - AA - AAA - 3FB

10 / ¿Cuáles son los máximos valores decimales que pueden expresarse con un número binario de:
2 cifras - 4 cifras - 8 cifras?

Las soluciones del test se encuentran en las págs. 76-77.

La revolución informática

La información, que es la base de la actividad intelectual del hombre, y la información elaborada –es decir, la informática– *están destinadas a jugar un papel cada vez más significativo en nuestra realidad más inmediata. Naturalmente, se trata de esa realidad, de ese esbozo de futuro que es posible indagar en la medida en que su desarrollo está predeterminado por las actividades de investigación de los últimos años. Y su grado de certidumbre, evidentemente, no es absoluto, aunque sí lo suficientemente elevado como para que podamos bosquejarlo.*
Es como si nos encontráramos ante un iceberg y describiéramos la parte visible: ésta nos puede dar una idea bastante exacta de la parte sumergida, que es mucho mayor. Y de este modo podemos afirmar que los ordenadores y la microelectrónica serán los protagonistas y los impulsores del cambio en las próximas décadas: la transformaciones más significativas que tendrán lugar ya tienen nombre y consistencia en el terreno del estudio y la experimentación. Sus grandes líneas conceptuales se identifican con los términos **telemática, robótica** *y* **burótica.**
La eclosión de la electrónica invadirá todos los planos de la vida cotidiana en la casa, en la oficina y en la fábrica. Dicho cambio no será automático, y se traducirá en un auténtico desarrollo sólo si la cultura evoluciona hacia un conocimiento generalizado, masivo, de la informática.

Telemática *es un término obtenido por la contracción de dos palabras:* telecomunicaciones *e* informática. *Se trata, pues, de la tecnología de las comunicaciones en conexión con la informática, que representa el primer gran hito del cambio.*
El teléfono, es decir, el terminal más difundido en el mundo (se calculaba en 500 millones el número de aparatos instalados a finales de 1980), el televisor y el ordenador personal que puede conectarlos, son las máquinas por las que pasará el cambio, que será posible gracias al perfeccionamiento de las comunicaciones: por cable, con la introducción de nuevos materiales (como, por ejemplo, las fibras ópticas); por el espacio, con la utilización a escala planetaria de los satélites artificiales. Una vez más la información digital, es decir, en código binario, *ya sea gráfica, alfabética o numérica, será el hito cualitativo de una nueva revolución que nos disponemos a vivir. El progreso se medirá por la velocidad de transmisión de la información, por la capacidad de almacenarla en «bancos de datos», recuperarla y transmitirla a grandes distancias.* Telemática *significará toda una serie de servicios canalizados por redes de telecomuni-*

El Videotel dispone de un teclado con acceso al banco de datos, y el televisor sirve de monitor.



SIP

caciones tradicionales o de nueva concepción, que volverán disponibles las más diversas informaciones contenidas en los bancos de datos, que son bloques de información organizados. Esto permite entrever la importantísima dinámica de desarrollo que la telemática aportará a diversos sectores y actividades industriales. Algunos sistemas tradicionales se modificarán y otros tendrán que ser creados como consecuencia y con el empuje de nuevas tecnologías. En primer lugar, la telemática actuará como factor de desarrollo ulterior para las organizaciones televisivas, telefónicas y de comunicaciones encargadas de proveer las estructuras que transmitirán, por cable o éter, la información a los usuarios. En segundo lugar, el desarrollo inducido por la telemática afectará a todas las estructuras dedicadas a crear, organizar y suministrar información. Las editoriales, que tradicionalmente juegan este papel, verán abrirse una ulterior salida sinérgica para la información. También otros organismos económicos, como, por ejemplo, las instituciones de crédito o la bolsa, encontrarán prometedoras vías de desarrollo en la telemática, que les permitirá distribuir mejor la información de que disponen. Tampoco la publicidad permanecerá ajena a esta nueva forma de comunicación, y la telemática representará también para ella un nuevo trámite para llegar al destinatario de la información. En última instancia, todo esto significa máquinas de los más diversos tipos: televisores de nueva concepción, ordenadores personales, teléfonos, instrumental para telecomunicaciones, etc., lo que significa nuevas posibilidades de desarrollo industrial.

Los robots han dejado de ser cosa de ciencia ficción. En la industria, en la investigación, en las actividades espaciales, los robots ya están en acción, sustituyendo a los hombres en las tareas repetitivas, duras y peligrosas. Su utilización permite reducir los costos de producción, hace posibles actividades en las que antes no cabía ni pensar, aumenta la eficacia de los procesos. En el futuro veremos cómo los robots mecánicos (ya hoy dotados de órganos sensoriales y dirigidos por dispositivos electrónicos y microprocesadores) se multiplican y se utilizan en las tareas más diversas.
La mayoría de las veces estas máquinas constan de un «brazo» mecánico que puede realizar todos los movimientos en todas las posiciones, y de una «mano» que sirve para coger, soltar,



Marka

El sistema Télex utiliza la red telefónica y está provisto de una impresora.

empuñar objetos; pero, cada vez con más frecuencia, los robots están dotados de órganos sensoriales que les permiten ver y tocar, o incluso hablar. Otra característica de los robots industriales es la de servir para muchos usos, con capacidad de memorizar las propias tareas y aprender otras nuevas. Lo que supone que, junto a una estructura mecánica, hay una estructura electrónica basada en un microprocesador y una memoria; esto es, a fin de cuentas, en un microordenador.
Los campos de utilización de los robots industriales son numerosos. Los encontramos sobre todo en la industria automovilística, pero también en otros sectores, donde realizan tareas imposibles o peligrosas para el hombre, duras o repetitivas y tediosas.

Una vez más, en la historia de la información, un término nace de la fusión de dos palabras y de dos conceptos, para indicar una relación empírica de concreción reciente, que define las técnicas y las tecnologías relativas a la automación del trabajo de oficina. El término es de acuñación francesa: Bureau, *es decir, oficina, e* informatique *se han fundido en* bureautique, *que nosotros traducimos como* burótica.

(Fragmento de EL ORDENADOR EN EL COLEGIO, de Egidio Pentiraro, Laterza, Bari, 1983.)

Suma binaria

Para efectuar sumas en aritmética binaria basta recordar que la base de dicha numeración es 2 y, por lo tanto, cuando el resultado de la adición es 2 escribimos un 0 y llevamos 1.

Exactamente el mismo procedimiento que seguimos en las sumas normales cuando el resultado es 10 o superior a 10.

Por ejemplo, sumemos el número binario 1 consigo mismo:

$$\begin{array}{r} 1\,+ \\ +1\,= \\ \hline 10 \end{array}$$

El resultado de 1 + 1 es 2; por lo tanto, escribimos un 0 y llevamos 1.

Para comprobar la operación, convirtamos todos los números en decimales:

$$\begin{array}{rl} & 1 \text{ binario} = 1 \text{ decimal} \\ & + \qquad\qquad + \\ & 1 \text{ binario} = 1 \text{ decimal} \\ \hline \text{Resultado:} & 10 \text{ binario} = 2 \text{ decimal} \end{array}$$

Sumemos ahora los dos números binarios:

1 0 1 1 0 1
1 0 1 1 0

Antes de efectuar la suma, convirtamos los dos números binarios en sus equivalentes decimales (utilizaremos estos valores para la comprobación del resultado):

1 0 1 1 0 1 $= 2^0 + 2^2 + 2^3 + 2^5 = 1 + 4 + 8 + 32 = 45$ decimal

1 0 1 1 0 $= 2^1 + 2^2 + 2^4 = 2 + 4 + 16 = 22$ decimal

luego el resultado de la suma binaria tendrá que ser equivalente a 45 + 22 = 67 decimal.

Examinemos la suma binaria: pongamos en columnas las cifras de los dos números a partir de la derecha:

Columna	g	f	e	d	c	b	a
1.er sumando		1	0	1	1	0	1
2.º sumando			1	0	1	1	0
	1	0	0	0	0	1	1

1 1 1 1
Llevamos

Empezando por la **columna A**: los dos sumandos son 1 y 0, el resultado es 1 y no llevamos nada.

Columna B: los sumandos son 0 y 1; resultado = 1; no llevamos nada.

Columna C: sumandos 1 y 1; resultado = 0; llevamos 1.

1 + 1 = 0 y llevamos 1.

Del mismo modo que en el sistema decimal tenemos:

8 + 2 = 10, o sea 0 y llevamos 1.

Lo que llevamos se sumará al resultado de la columna D.

Columna D: sumandos 1 y 0; resultado = 1; sumándole lo que llevábamos (1): 1 + 1 = nuevo resultado = 0. Y de nuevo llevamos 1.

Columna E: sumandos 0 y 1; resultado = 1; sumamos lo que llevábamos (1): 1 + 1 = nuevo resultado = 0. Llevamos 1.

Columna F: sumandos 1 y 0; resultado = 1; sumamos lo que llevábamos (1): 1 + 1 = nuevo resultado = 0. Llevamos 1.

Columna G: no hay sumandos, luego se anota lo que llevábamos de la columna anterior (1):

1 0 0 0 0 1 1 $= 2^0 + 2^1 + 2^6 = 1 + 2 + 64 = 67$ decimal.

De forma totalmente análoga se pueden efectuar las sumas en base 8 y en base 16.

En estas notaciones hay que recordar que llevamos algo cuando la suma es, respectivamente, igual (o mayor) que 8 y 16.

Por ejemplo, efectuemos la suma de los mismos dos números, pero empleando la notación octal. Pasemos los dos sumandos a la notación octal:

valor	4	2	1		4	2	1
binario a transformar	1	0	1		1	0	1
correspondiente octal	5 (4+1)				5 (4+1)		

valor	4	2	1		4	2	1
binario a transformar	*	1	0		1	1	0
correspondiente octal	2				6		

1 0 1 1 0 1 = 55 octal
1 0 1 1 0 = 26 octal

* La ausencia de símbolos en las primeras posiciones equivale a poner un 0. Así, el número binario 1 0 1 1 0 es equivalente a 0 1 0 1 1 0.

La suma, en octal, será:

	5	5	octal +
	2	6	octal =
1	0	3	
	1	1	

El resultado (103 octal) se obtiene del modo siguiente:

- Empezando por la derecha, $5+6=11$; por tanto (quitando la base queda 3 y llevamos 1) en la segunda columna tendremos: $5+2+1$ que llevamos $=8$; por lo tanto, quitando la base nos queda 0 y llevamos 1.

En la última columna se escribe sólo lo que llevamos, pues no hay sumandos.

Pasemos el resultado al sistema decimal:

El número
103 octal significa:

3 «unidades octales» $= 3\times 8^0 = 3$ +
0 «decenas octales» $= 0\times 8 = 0$ +
1 «centena octal» $= 1\times 64 = 64$

Sumando 67

De forma más sintética:
103 octal $= 3\times 8^0 + 1\times 8^2 = 3 + 64 = 67$

Diferencia binaria

La diferencia entre dos números en notación binaria puede efectuarse con las mismas reglas de la diferencia decimal; pero, por motivos que veremos con más detalle, utilizaremos el «método del complemento».

El complemento de un número binario se obtiene sustituyendo todos los 1 por 0 y viceversa. Así, el complemento de 1 0 0 1 0 1 es 0 1 1 0 1 0.

La suma de un número y su complemento es un número binario en el que todas las cifras son 1; en efecto:

número	**1**	**0**	**0**	**1**	**0**	**1**	+
complemento	0	1	1	0	1	0	
suma	1	1	1	1	1	1	

Si, además, se añade 1 a la suma, todos los 1 se convierten en 0: 1 1 1 1 1 1 + 1 = 1 0 0 0 0 0 0.

Con el método del complemento, la diferencia entre dos números se obtiene así:

A / Se complementa el sustraendo poniéndole delante los 0 necesarios para que tenga el mismo número de cifras que el minuendo.

B / Si efectúa la suma del minuendo con el complemento del sustraendo, eliminando del resultado el eventual 1 que puede rebasar la longitud del minuendo.

C / Se añade 1.

Por ejemplo:
se quiere efectuar la resta:
minuendo sustraendo
1 0 0 0 1 1 – 1 0 1 0 1
En decimal la diferencia es: $35-21=14$.

Paso A:

El sustraendo tiene un número de símbolos menor que el del minuendo.
Se completa con ceros, y la operación se plantea de la forma siguiente:

Minuendo	1	0	0	0	1	1
Sustraendo	0	1	0	1	0	1
Se halla el complemento del sustraendo =	1	0	1	0	1	0

Paso B:

Se efectúa la suma del minuendo (100011) con el complemento del sustraendo (101010):

minuendo		**100011**
complemento del sustraendo		**101010**
suma	**1**	**001101**

Se excluye de la suma el primer 1, puesto que excede la longitud del minuendo.

Paso C:

Se añade 1 a la suma:

suma	**001101** +
	1
resultado	**001110** = 14 decimal

El procedimiento, tal como ha sido descrito, puede parecer mucho más engorroso de lo que es realmente. En efecto, con un mínimo de

práctica, muchos de los pasos que en el ejemplo son explicitados, pueden realizarse mentalmente.

Pongamos otro ejemplo, exponiéndolo de forma sintética.

Desarrollemos la operación:

226 − 174 = 52 decimal =

= **11100010 − 10101110**

Complementando el sustraendo (**10101110**) y efectuando la suma, tenemos:

minuendo		**11100010**
complemento sustraendo		**01010001**
resultado suma	**1**	**00110011** +
		1
		00110100 =
		= 52 decimal

Para concluir, en la tabla 1 (págs. 63 a 66) tenemos una lista parcial de las equivalencias decimal-binario-octal-hexadecimal de los números 1 a 255.

En la tabla 2 (págs. 66 y 67) se dan las mismas equivalencias para los números comprendidos entre 31100 y 31150.

Como veremos, los datos de las columnas OCT (octal) y HEX (hexadecimal) han sido obtenidos por un ordenador con una sola instrucción que da el resultado en milésimas de segundo.

Intente traducir, por ejemplo, el número 31142; aunque sea muy rápido, tardará algunos minutos.

Ordenador HP 9000 que muestra el curso de las señales digitales de un circuito lógico.



Hewlett-Packard

TABLA 1

DECIMAL	BINARIO	OCT	HEX
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	1	1
2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	2	2
3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1	3	3
4	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	4	4
5	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	5	5
6	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0	6	6
7	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1	7	7
8	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	10	8
9	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1	11	9
10	0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0	12	A
11	0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1	13	B
12	0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0	14	C
13	0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1	15	D
14	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0	16	E
15	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	17	F
16	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0	20	10
17	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1	21	11
18	0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0	22	12
19	0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1	23	13
20	0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0	24	14
21	0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1	25	15
22	0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0	26	16
23	0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1	27	17
24	0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0	30	18
25	0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1	31	19
26	0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0	32	1A
27	0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1	33	1B
28	0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0	34	1C
29	0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1	35	1D
30	0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0	36	1E
31	0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1	37	1F
32	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	40	20
33	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1	41	21
34	0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0	42	22

DECIMAL	BINARIO	OCT	HEX
35	0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1	43	23
36	0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0	44	24
37	0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1	45	25
38	0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0	46	26
39	0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1	47	27
40	0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0	50	28
41	0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1	51	29
42	0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0	52	2A
43	0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1	53	2B
44	0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0	54	2C
45	0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1	55	2D
46	0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0	56	2E
47	0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1	57	2F
48	0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0	60	30
49	0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1	61	31
50	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0	62	32
51	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1	63	33
52	0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0	64	34
53	0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1	65	35
54	0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0	66	36
55	0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1	67	37
56	0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0	70	38
57	0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1	71	39
58	0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0	72	3A
59	0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1	73	3B
60	0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0	74	3C
61	0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1	75	3D
62	0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0	76	3E
63	0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1	77	3F
64	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	100	40
65	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1	101	41
66	0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0	102	42
67	0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1	103	43
68	0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0	104	44
69	0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1	105	45

215	0 0 0 0 0 0 0 0 1 1 0 1 0 1 1 1	327	D7
216	0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0	330	D8
217	0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 1	331	D9
218	0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0	332	DA
219	0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 1	333	DB
220	0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0	334	DC
221	0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 1	335	DD
222	0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0	336	DE
223	0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1	337	DF
224	0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0	340	E0
225	0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1	341	E1
226	0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0	342	E2
227	0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1	343	E3
228	0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0	344	E4
229	0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1	345	E5
230	0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0	346	E6
231	0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1	347	E7
232	0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0	350	E8
233	0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1	351	E9
234	0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0	352	EA
235	0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 1	353	EB
236	0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 0	354	EC
237	0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 1	355	ED
238	0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0	356	EE
239	0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1	357	EF
240	0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0	360	F0
241	0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1	361	F1
242	0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 0	362	F2
243	0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1	363	F3
244	0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0	364	F4
245	0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1	365	F5
246	0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0	366	F6
247	0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1	367	F7

DECIMAL	BINARIO	OCT	HEX
248	0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0	370	F8
249	0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1	371	F9
250	0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0	372	FA
251	0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1	373	FB
252	0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0	374	FC
253	0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1	375	FD
254	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0	376	FE
255	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1	377	FF

TABLA 2

DECIMAL	BINARIO	OCT	HEX
31100	0 1 1 1 1 0 0 1 0 1 1 1 1 1 0 0	74574	797C
31101	0 1 1 1 1 0 0 1 0 1 1 1 1 1 0 1	74575	797D
31102	0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 0	74576	797E
31103	0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1	74577	797F
31104	0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0	74600	7980
31105	0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1	74601	7981
31106	0 1 1 1 1 0 0 1 1 0 0 0 0 0 1 0	74602	7982
31107	0 1 1 1 1 0 0 1 1 0 0 0 0 0 1 1	74603	7983
31108	0 1 1 1 1 0 0 1 1 0 0 0 0 1 0 0	74604	7984
31109	0 1 1 1 1 0 0 1 1 0 0 0 0 1 0 1	74605	7985
31110	0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 0	74606	7986
31111	0 1 1 1 1 0 0 1 1 0 0 0 0 1 1 1	74607	7987
31112	0 1 1 1 1 0 0 1 1 0 0 0 1 0 0 0	74610	7988
31113	0 1 1 1 1 0 0 1 1 0 0 0 1 0 0 1	74611	7989
31114	0 1 1 1 1 0 0 1 1 0 0 0 1 0 1 0	74612	798A
31115	0 1 1 1 1 0 0 1 1 0 0 0 1 0 1 1	74613	798B

31116	0	1	1	1	1	0	0	1	1	0	0	0	1	1	0	0	74614	798C
31117	0	1	1	1	1	0	0	1	1	0	0	0	1	1	0	1	74615	798D
31118	0	1	1	1	1	0	0	1	1	0	0	0	1	1	1	0	74616	798E
31119	0	1	1	1	1	0	0	1	1	0	0	0	1	1	1	1	74617	798F
31120	0	1	1	1	1	0	0	1	1	0	0	1	0	0	0	0	74620	7990
31121	0	1	1	1	1	0	0	1	1	0	0	1	0	0	0	1	74621	7991
31122	0	1	1	1	1	0	0	1	1	0	0	1	0	0	1	0	74622	7992
31123	0	1	1	1	1	0	0	1	1	0	0	1	0	0	1	1	74623	7993
31124	0	1	1	1	1	0	0	1	1	0	0	1	0	1	0	0	74624	7994
31125	0	1	1	1	1	0	0	1	1	0	0	1	0	1	0	1	74625	7995
31126	0	1	1	1	1	0	0	1	1	0	0	1	0	1	1	0	74626	7996
31127	0	1	1	1	1	0	0	1	1	0	0	1	0	1	1	1	74627	7997
31128	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	74630	7998
31129	0	1	1	1	1	0	0	1	1	0	0	1	1	0	0	1	74631	7999
31130	0	1	1	1	1	0	0	1	1	0	0	1	1	0	1	0	74632	799A
31131	0	1	1	1	1	0	0	1	1	0	0	1	1	0	1	1	74633	799B
31132	0	1	1	1	1	0	0	1	1	0	0	1	1	1	0	0	74634	799C
31133	0	1	1	1	1	0	0	1	1	0	0	1	1	1	0	1	74635	799D
31134	0	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	74636	799E
31135	0	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1	74637	799F
31136	0	1	1	1	1	0	0	1	1	0	1	0	0	0	0	0	74640	79A0
31137	0	1	1	1	1	0	0	1	1	0	1	0	0	0	0	1	74641	79A1
31138	0	1	1	1	1	0	0	1	1	0	1	0	0	0	1	0	74642	79A2
31139	0	1	1	1	1	0	0	1	1	0	1	0	0	0	1	1	74643	79A3
31140	0	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	74644	79A4
31141	0	1	1	1	1	0	0	1	1	0	1	0	0	1	0	1	74645	79A5
31142	0	1	1	1	1	0	0	1	1	0	1	0	0	1	1	0	74646	79A6
31143	0	1	1	1	1	0	0	1	1	0	1	0	0	1	1	1	74647	79A7
31144	0	1	1	1	1	0	0	1	1	0	1	0	1	0	0	0	74650	79A8
31145	0	1	1	1	1	0	0	1	1	0	1	0	1	0	0	1	74651	79A9
31146	0	1	1	1	1	0	0	1	1	0	1	0	1	0	1	0	74652	79AA
31147	0	1	1	1	1	0	0	1	1	0	1	0	1	0	1	1	74653	79AB
31148	0	1	1	1	1	0	0	1	1	0	1	0	1	1	0	0	74654	79AC
31149	0	1	1	1	1	0	0	1	1	0	1	0	1	1	0	1	74655	79AD
31150	0	1	1	1	1	0	0	1	1	0	1	0	1	1	1	0	74656	79AE

Un terminal en el televisor

Una pantalla de televisión normal puede hoy día convertirse en el terminal de un sofisticado ordenador electrónico, permitiendo a cualquiera insertarse como usuario en una red de transmisión de datos.

Desde hace ya algún tiempo, podemos tener acceso a una limitada serie de informaciones de carácter general, como por ejemplo las relativas a las condiciones y previsiones meteorológicas, la hora exacta, el estado de carreteras, etc., simplemente marcando ciertos números de teléfono. El salto cualitativo, sin embargo, se ha dado sólo recientemente, con la introducción de los sistemas combinados de elaboración y difusión de la información. Hoy día, muchos usuarios de la televisión, pulsando una determinada combinación de teclas del mando a distancia, pueden obtener en su pantalla (en sobreimpresión o a todo campo) una amplia gama de fichas informativas relativas a los temas más diversos. Actualmente hay disponibles, a nivel operativo, dos sistemas combinados que permiten la prestación de este tipo de servicio: el Teletext *y el* Viewdata. *El primero, de los años setenta, se difunde a través de la red de televisión normal, mientras que el segundo se transmite mediante la red telefónica, y un sistema de decodificación especial permite visualizar informaciones en la pantalla del usuario.*

El Teletext *aprovecha los «retales» de la pequeña pantalla que permanecen inutilizados a lo largo de los programas de televisión. Mientras el usuario está mirando un programa cualquiera puede, pulsando un determinado botón del mando a distancia, hacer que aparezca en la pantalla la primera página de la serie de informaciones* Teletext *disponible en ese momento. La primera página contiene el índice general de la serie completa, y basta con marcar en el teclado el número de la página que interesa, para*

El Prestel envía informaciones de diversos tipos a través de las líneas telefónicas.



Standard Telephones & Cables Ltd.

que ésta aparezca en la pantalla, donde permanece hasta que no se marque otro número.
Las imágenes del Teletext son preparadas por pequeños grupos editoriales, que insertan regularmente en el ordenador central las necesarias informaciones. Cada página de transmisión está formada por 24 líneas de texto, de 40 caracteres cada una. En la composición del texto se pueden utilizar siete colores (blanco, rojo, azul, verde, magenta, celeste y amarillo). Cada signo se transmite utilizando un código numérico capaz de especificar a los circuitos electrónicos destinados a la formación de la imagen en el aparato receptor, cuáles de los 70 puntos que constituyen en la pantalla el «paquete» destinado a acoger un signo, deben iluminarse. Una segunda información codificada especifica el color de cada «paquete». Un técnico describiría con más propiedad el mismo mecanismo, sustituyendo indignado el término «paquete» por el de campo destinado en la pantalla a cada signo, implementable en una matriz de 70 pixel. Algunas posibilidades gráficas permiten, además, unificar los campos correspondientes a varios signos para formar letras y títulos mayores, además de colorear de manera selectiva algunos campos para conseguir imágenes sencillas en dos dimensiones.
Durante el horario de transmisión, que generalmente coincide con el de los programas de televisión normales, la secuencia completa de las páginas del text se emite cíclicamente sin pausa. Una página de 24 líneas impresas requiere, para ser transmitida, algo menos de un cuarto de segundo: un intervalo de tiempo nada despreciable. Si se tiene en cuenta que un servicio de Teletext normal nunca consta de menos de cien páginas, y que todo el servicio se ha de transmitir cíclicamente línea a línea, página a página, desde el momento en que el usuario pide una determinada página pueden pasar varios minutos antes de que ésta se forme en la pantalla. El «black-out» de la imagen se evita, mientras tanto, gracias a un circuito de memoria que retiene en la pantalla la página anteriormente seleccionada hasta que no llega la que ha sido pedida. Este tiempo muerto de espera constituye uno de los principales inconvenientes del Teletext, y ya se ha previsto solucionarlo.
Dos perfeccionamientos del Teletext, actualmente en fase experimental, permiten prever la posibilidad de un notable desarrollo de sus campos de aplicación. El primer proyecto se denomina Telesoftware: según los técnicos, la incorporación de un microprocesador al circuito electrónico receptor permitirá, en el futuro, tener acceso directo al ordenador central o a una red de ordenadores interconectados. De este modo, se podrá utilizar el televisor como un «terminal inteligente», capaz de «efectuar» programas de cálculo adecuados a nuestras necesidades concretas, como, por ejemplo, la contabilidad y la gestión de la economía doméstica.
El segundo proyecto atañe a otro servicio obtenible en el ámbito del sistema Teletext, consistente en añadir, por petición expresa del telespectador, subtítulos a los programas de TV normales. Por una parte, ello permitiría subsanar las grandes dificultades afrontadas por las personas que padecen de trastornos auditivos; por otra parte, cualquiera tendría la posibilidad de seguir los programas en lengua extranjera en la misma lengua de origen.
El Viewdata es el sistema alternativo al Teletext, y ofrece mayores posibilidades a largo plazo. Conocido con otras denominaciones, distintas en los varios países en que ha sido introducido (Videotext, Videotel, Prestel, Televídeo), el sistema conecta los millares de «terminales televisivos» (puesto que ya se pueden llamar así) con el ordenador central mediante las líneas telefónicas normales. El Prestel, por ejemplo, es el servicio Viewdata ofrecido por el instituto británico de telecomunicaciones, y ha sido el primero que se ha establecido en Europa. Para tener acceso al servicio, el usuario, tras haber encendido el televisor, pulsa un botón especial, exactamente igual que en el caso del Teletext; el televisor se prepara para la nueva función y se conecta automáticamente con el ordenador central a través del aparato telefónico de la casa. En otras versiones menos avanzadas hay que efectuar «manualmente» la operación de conexión con la central de transmisión, lo que se consigue, simplemente, marcando en el teléfono el número de la central. Entonces se obtiene el índice general de las informaciones difundidas por el servicio, se selecciona en el índice el tema que interesa y se compone en el teclado del mando a distancia el número-clave que da acceso a esa página. En la pantalla aparece entonces un segundo índice analítico más detallado, y la selección de los temas puede seguir en base a esta ramificación de datos que el servicio ofrece mediante una sucesión de selecciones en forma de árbol lógico.



SIP

Algunas de las «caretas» ofrecidas en Italia por el SIP con el Videotel.

Las posibilidades de un sistema Viewdata *son enormes. A diferencia de lo que ocurre con el* Teletext, *las distintas páginas no son transmitidas por la línea del usuario si éste no las solicita expresamente. Ello permite el almacenamiento de una gran cantidad de informaciones que pueden ser transmitidas sin dilación al ser solicitadas. Los límites de un sistema de este tipo no son otros, en esencia, que los de la capacidad de memoria del ordenador central.*

El sistema Prestel *recibe los datos que lo mantienen continuamente actualizado en 150 organismos especializados en distintos sectores que «producen información», y actualmente está en condiciones de suministrar a los usuarios datos sobre los temas más diversos: tiempo libre, viajes y medios de transporte, precios del mercado, estadísticas, etc. El usuario paga las páginas de información que ha ido solicitando; el precio de cada página aparece en la telepantalla junto con la página misma, y el ordenador registra automáticamente el importe en la cuenta del usuario. Éste puede, en cualquier momento, pedir al ordenador un extracto de su cuenta, que aparece directamente en la pantalla.*

En un futuro próximo, un sistema Viewdata *podría permitir el acceso mediante línea a los bancos de datos disponibles a quienquiera que estuviera interesado, por razones profesionales o culturales, en la adquisición de determinados tipos de información. Sociedades como la Lockheed y grupos universitarios como el Instituto Tecnológico de Massachusetts, ofrecen ya amplios bancos de datos a los particulares.*

Es evidente que un servicio de esta magnitud estaría orientado preferentemente hacia el mundo profesional y comercial. Una interesante característica de este servicio es la posibilidad de acceso limitado, gracias a la cual algunas páginas del mismo sólo serían accesibles a determi-

nadas entidades o personas, permitiendo, entre otras cosas, la rápida comunicación de información reservada. Puede conseguirse un alto nivel de reserva asignando a cada usuario una sigla alfanumérica personal, que funciona como «palabra clave» (password) para el ordenador, que puede permitir o no el acceso de un determinado usuario a una sección del archivo que contiene informaciones reservadas. El peligro de que las informaciones transmitidas por la línea de conexión del usuario sean interceptadas por terceros se puede eliminar también adoptando códigos de transmisión especiales, que hacen que las informaciones sólo resulten comprensibles para el usuario al que están destinadas. Esto, sin embargo, exige la integración en el sistema receptor de un microprocesador programado para decodificar el mensaje recibido.
Otra característica peculiar del sistema Viewdata la constituye el hecho de que el usuario puede comunicar directamente con el ordenador central de forma interactiva, y ello permite entrever fácilmente el alcance de los posibles desarrollos. Un servicio que requiere el contacto interactivo entre el usuario y el ordenador es, por ejemplo, el diagnóstico médico computarizado. Los programas de diagnóstico ya en uso en los más importantes hospitales ahorran a los médicos un tiempo precioso en el análisis de las condiciones de los pacientes que llegan, y un servicio análogo en casa, utilizado con las debidas precauciones, podría ahorrar mucho trabajo al médico de cabecera.
Para introducir un grado de interactividad mayor, haría falta dotar al sistema receptor de un microprocesador y un teclado alfanumérico. Ello permitiría al usuario el acceso, no sólo a los archivos de masa del ordenador, sino también a la mismísima unidad central de proceso de datos, con objeto, por ejemplo, de hacerle realizar programas personales y obtener los resultados en su propia pantalla. El costo de un servicio de este tipo, sobre todo si se utilizara sin demasiada frecuencia y con programas sencillos, podría resultar inferior al de un ordenador personal.
Otra posibilidad ofrecida por la incorporación de un microprocesador, es la de conectarse con el propio banco usando como «puente» el ordenador que dirige el sistema de difusión, para llevar a cabo cualquier operación, incluso la transferencia de fondos a o desde la propia cuenta corriente. El televisor casero funciona, en este caso, como uno de los terminales de vídeo de la filial bancaria, permitiendo la realización de las operaciones habituales sin moverse de casa y en tiempo real.

La presentación de la información ferroviaria con el servicio Videotel.



Elementos de lógica

Algunas de las funciones que realiza un ordenador pueden ser obtenidas también con otros tipos de circuitos. En este capítulo se tratarán los aspectos fundamentales de esta técnica, que se denomina **lógica cableada**.

El tema no atañe directamente a los ordenadores, pero constituye el primer paso indispensable para comprender la estructura y el funcionamiento de estas máquinas.

Empecemos viendo algunos conceptos de lógica que serán retomados y ampliados en la parte dedicada a los lenguajes de programación.

Los operadores

Un microordenador, o más exactamente un componente suyo: la CPU (Central Processing Unit) puede realizar, además de operaciones aritméticas, **operaciones lógicas**.

Estas se utilizan principalmente para efectuar comparaciones entre datos y activar, en función del resultado, una determinada acción.

Por ejemplo, en un programa que calcule los sueldos se puede insertar un control sobre los nombres que se introducen; si dicho nombre es FIN, la máquina tiene que avisar al operador y terminar el programa.

En este caso la operación lógica es la comparación entre el nombre que se introduce y la palabra FIN. Si el resultado es negativo (si el nombre no es FIN), la máquina sigue funcionando; si el resultado es positivo (si el nombre es FIN), la máquina emite una señal y se para.

Los operadores lógicos pueden considerarse, en algunos aspectos, equivalentes a los operadores aritméticos normales; por lo tanto, introduciremos el tema a partir de esta analogía.

Al efectuar un cálculo aritmético normal, podemos distinguir tres elementos fundamentales:

- **los operandos**
- **el operador**
- **el resultado**

Los operandos son los números con los que se realiza la operación; el operador es el símbolo que nos indica qué operación hay que efectuar, y el resultado es el número que se asocia, por medio del operador, a los operandos.

Consideremos, por ejemplo, la operación aritmética $5+3=8$.

El resultado (8) se asocia a los operandos 5 y 3 mediante el operador + (suma).

Si cambiásemos el elemento operador (por ejemplo −, resta), los símbolos 5 y 3 se asociarían (mediante este nuevo operador) al número decimal 2. Las distintas operaciones (suma, resta, producto, etc.) son **leyes asociativas**, es decir, leyes que asocian grupos de números (operandos) a otros números (resultados); los tipos de asociación se expresan mediante símbolos operadores (+, −, ×, etc.).

Este ejemplo entraña una regla fundamental: una ley asociativa (por ejemplo, la suma, la resta, etc.) sólo puede aplicarse entre elementos homogéneos, es decir, pertenecientes al mismo conjunto (ver gráfico). Sin adentrarnos en temas que exceden la intención de este capítulo, intentemos aclarar el concepto de **conjunto**.



A nivel muy intuitivo, se puede entender por conjunto un grupo de elementos (números, letras, objetos) que satisfacen ciertas leyes.

Especificar cuáles son estas leyes y cuáles las propiedades de los elementos es tarea del álgebra abstracta. En este contexto se entiende por conjunto una agrupación de elementos homogéneos.

Este concepto de conjunto, aunque inexacto matemáticamente hablando, debe tenerse en cuenta en su aplicación a los ordenadores.

Al efectuar cálculos, muchos errores se deben a la utilización de operandos no homogéneos. (Este tema será tratado minuciosamente, desde el punto de vista operativo, más adelante.)

Nos hemos referido explícitamente a los operadores de las operaciones aritméticas normales, pero nada nos impide inventar una ley asociativa cualquiera, darle un nombre y utilizarla como un nuevo operador.

En realidad el procedimiento no es tan simplista; el objetivo de este comentario no es otro que el de introducir un nuevo tipo de operadores: los **operadores lógicos.**

Dichos operadores tienen sus propias reglas, como los operadores aritméticos (+, –, etc.), y cumplen una serie de leyes, a menudo muy complejas. En esta obra nos ocuparemos sólo de los aspectos que conciernen directamente a la programación de ordenadores.

Los operadores lógicos

Volvamos al ejemplo del semáforo; éste puede hallarse en uno de los estados siguientes:

1 - apagado
2 - rojo
3 - ámbar
4 - verde

A la pregunta «¿cómo puede hallarse un semáforo?», hemos de responder que sólo puede hallarse en uno de dichos estados.

En otras palabras, cualquiera de las situaciones posibles excluye las demás.

Por tanto, nuestra respuesta a la pregunta tendrá que ser:

apagado o bien **rojo** o bien **ámbar** o bien **verde.**

La expresión «o bien» es un operador lógico denominado **OR.**

La respuesta anterior, indicando los estados posibles con el número correspondiente (1 = apagado, 2 = rojo, etc.), se convierte en:

estado del semáforo = 1 OR 2 OR 3 OR 4

Supongamos ahora que un automóvil llegue ante el semáforo: ¿cuándo puede pasar?

Las posibilidades son sólo dos: **semáforo apagado** o bien **semáforo verde.**

Expresando la respuesta en forma sintética:

ADELANTE = 1 OR 4 (1 = apagado; 4 = verde)

En realidad, la expresión 1 OR 4 no es suficiente, puesto que si el semáforo está apagado hay que cerciorarse de que no haya otros vehículos que impidan el paso; por tanto, la respuesta completa es:

ADELANTE = **semáforo verde** o bien **semáforo apagado** y **vía libre**

La conjunción «y» significa que las condiciones semáforo apagado, vía libre deben cumplirse simultáneamente; es un nuevo operador lógico, denominado **AND.**

Codificando con el número 5 la situación de vía libre, la condición anterior se convierte en:

ADELANTE = 4 OR 1 AND 5

Esta expresión lógica, tal como está escrita, puede interpretarse de dos maneras:

1 - ADELANTE = (4 OR 1) AND 5
2 - ADELANTE = 4 OR (1 AND 5)

La primera significa que se puede pasar al cumplirse 4 o 1 (4 OR 1) a la vez que (AND) 5. Es, por tanto, una interpretación errónea.

La segunda significa que se puede pasar si se cumple 4, o bien (OR) las condiciones simultáneas (AND) 1 y 5. Esta es la interpretación correcta.

En este ejemplo se ve claramente la necesidad de usar de forma correcta los paréntesis, igual que en las operaciones aritméticas normales.

Los operadores AND y OR pueden aplicarse a dos elementos de un conjunto cualquiera. Para concretar, consideremos su aplicación a dos cifras binarias A y B.

Cada uno de estos dos elementos, al ser una cifra binaria, puede tener el valor 0 o el valor 1; por lo tanto, las posibles combinaciones entre los valores de A y B serán las siguientes:

A	B
1	1
1	0
0	1
0	0

Recordando el concepto de AND = a la vez, OR = o bien, tenemos:

1 A	2 B	3 A AND B	4 A OR B
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

Las columnas 1 y 2 (A y B) son los operandos.

La columna 3 es el resultado de la operación A AND B; dicho resultado es 1 cuando tanto A como B valen **1.**

La columna 4 es el resultado de la operación A OR B, y vale 1 cuando A o B (al menos uno de los dos) valen **1.**

Introduzcamos ahora el nuevo operador **XOR,** denominado también OR EXCLUSIVO.

Este operador tiene el mismo significado que OR excepto en el caso A = **1** y B = **1,** en el que su resultado es **0** (al contrario de OR, que da **1**).

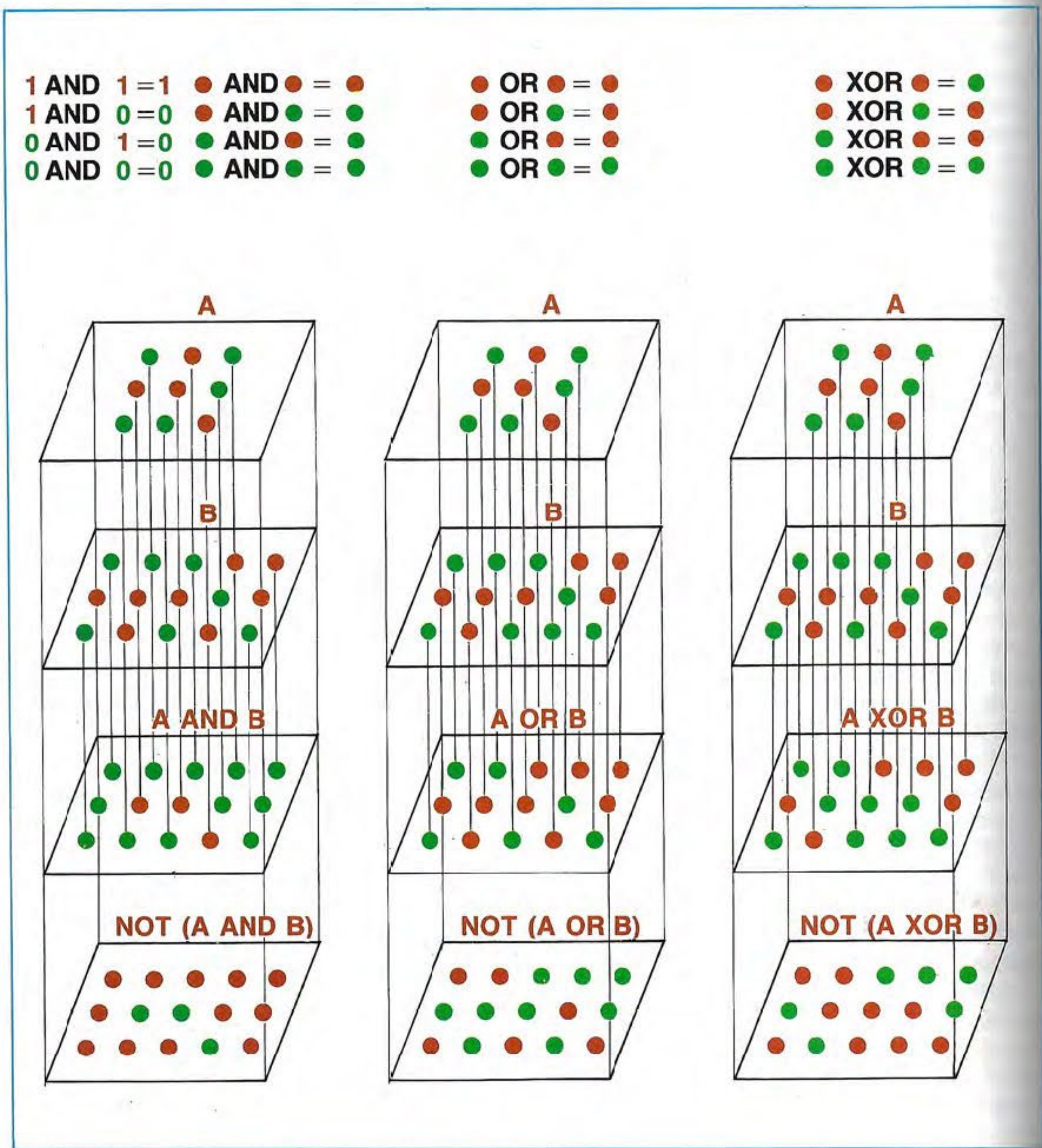
En otras palabras, XOR vale **1** cuando solamente uno de los elementos A y B vale **1.**
Por lo tanto, la **TABLA DE LA VERDAD** ce este operador es:

A	B	A XOR B
1	1	0
1	0	1
0	1	1
0	0	0

Por último, tenemos el operador **NOT;** se aplica a un solo operando y tiene el mismo significado que el complemento, es decir, sustituye el símbolo **1** por el **0** y viceversa:

A	NOT A
1	0
0	1

En el gráfico inferior (punto rojo = símbolo **1**; punto verde = símbolo **0**) se representan los ejemplos de aplicación de los operadores AND, OR, XOR y NOT a dos conjuntos binarios.



Resumiendo, los principales operadores lógicos son: AND, OR, XOR y NOT. Su TABLA DE LA VERDAD es:

A	B	AND	OR	XOR	NOT A	NOT B
1	1	1	1	0	0	0
1	0	0	1	1	0	1
0	1	0	1	1	1	0
0	0	0	0	0	1	1

Los operadores lógicos se rigen por las mismas reglas que los operadores aritméticos.
Las operaciones lógicas sólo se pueden efectuar entre elementos homogéneos.

Hasta ahora hemos aplicado los operadores lógicos a números binarios de una sola cifra (A = 1 /0; B = 1 /0); del mismo modo, se pueden aplicar los operadores lógicos a los números binarios de varias cifras.
Por ejemplo, tomemos los números:

A = 0 1 1 0 1
B = 1 0 0 1 1

Tendremos:

NOT A = 1 0 0 1 0
NOT B = 0 1 1 0 0

A AND B

A	0	1	1	0	1
B	1	0	0	1	1
A AND B	0	0	0	0	1

Los dos números binarios A y B, traducidos al sistema decimal, tienen los valores:

A = 0 1 1 0 1 = 13 decimal ($2^3 + 2^2 + 2^0$)
B = 1 0 0 1 1 = 19 decimal ($2^4 + 2^1 + 2^0$)

y puesto que A AND B = 1 (ejemplo anterior), el mismo resultado ha de ser válido si consideramos los valores decimales de A y B:

0 1 1 0 1 AND 1 0 0 1 1 = 13 AND 19 = 1

A OR B

A	0	1	1	0	1
B	1	0	0	1	1
A OR B	1	1	1	1	1

En sistema decimal, 13 OR 19 = 31 (1 1 1 1 1 = = $2^4 + 2^3 + 2^2 + 2^1 + 2^0$ = 31).

A XOR B

A	0	1	1	0	1
B	1	0	0	1	1
A XOR B	1	1	1	1	0

En decimal, 13 XOR 19 = 30 (1 1 1 1 0 = 30).

Operadora trabajando. El ordenador con el que opera es del tipo de teclado separado.



Marka

SOLUCIONES DEL TEST 1

1 / a - La notación binaria es una consecuencia del funcionamiento de los circuitos digitales. Dichos circuitos sólo pueden reconocer dos situaciones (ON = presencia de señal y OFF = ausencia); por lo tanto, sólo pueden utilizar el sistema
de numeración binario.

b - La notación octal permite expresar tres símbolos binarios con un solo símbolo octal; por ejemplo, el número binario 111 en octal es 7, el binario 101111 en octal es 57. El sistema octal es más compacto y fácil de usar.

c - Con una sola cifra hexadecimal se puede expresar, como máximo, el valor 15 (15 decimal = F hexadecimal), dado que una cifra hexadecimal tiene por base 16 y equivale a cuatro cifras binarias (1111 binario = F hexadecimal = 15 decimal). Para escribir números mayores de 15 hacen falta más cifras hexadecimales. Por ejemplo, el número decimal 27 en binario es 11011 y en hexadecimal es 1B.

d - La notación octal tiene como base el número decimal 8, por lo que dicho valor se expresa mediante el número octal 10.

---

2 / El paso del sistema binario al decimal es muy sencillo si se utiliza la tabla de potencias. En un número binario, la presencia del símbolo 1 indica que se ha de tomar la potencia de 2 correspondiente; escribiendo una tabla de potencias, el paso de binario a decimal consiste, simplemente, en sumar las potencias a las que corresponde un 1:

Potencia	2^5	2^4	2^3	2^2	2^1	2^0		
Valor	32	16	8	4	2	1		
			1	0	0	1	1 + 8 = 9	1001 binario = 9 dec.
					1	1	1 + 2 = 3	11 binario = 3 dec.
						1	1 =	1binario = 1 dec.

Los otros valores pueden comprobarse utilizando las tablas de conversión decimal-binario.

---

3 / La traducción binario-octal es inmediata dividiendo el número binario en grupos de tres cifras:

Binario	1 1 0	1 1 1
Octal	6	7

Binario		Octal
11	⟶	3
101010	⟶	52
111111	⟶	77

4 / En este caso el número binario se ha de dividir en grupos de cuatro cifras:

Binario		Hexadecimal
1 1 0 1 1 1 1 1	=	D F

Los demás valores se pueden comprobar en las tablas 1 y 2 (págs. 63-67).

5 / La conversión se efectúa tomando los restos de las divisiones por 2:

10	0
5	1
2	0
1	1

10 decimal = 1 0 1 0

100	0
50	0
25	1
12	0
6	0
3	1
1	1

100 decimal = 1 1 0 0 1 0 0

Análogamente, 1000 decimal = 1111101000 binario = 3E8 hexadecimal.

6 /

Binario	Decimal
0 0 0 1	1
0 0 1 0	2
0 1 0 0	4
1 0 0 0	8

Como puede verse, en un número binario el desplazamiento a la izquierda de una posición equivale a multiplicar por 2. Análogamente, un desplazamiento a la derecha equivale a una división por 2. A esta conclusión se llega también observando que en cualquier sistema de numeración posicional un desplazamiento a la izquierda equivale a multiplicar por la base. En la «jerga» de la programación, un desplazamiento (a derecha o izquierda) se denomina ROTACIÓN.

7 / a - cierto; b - cierto; c - falso; d - cierto.

8 / Para pasar un número octal al sistema decimal hay que multiplicar cada cifra octal por la correspondiente potencia de 8. A modo de ejemplo, efectuemos la conversión del número octal 700:

$$700 \text{ octal} = 7 \times 8^2 + 0 \times 8^1 + 0 \times 8^0 = 7 \times 64 + 0 + 0 = 448$$

9 / La conversión de un número hexadecimal es más sencilla si se pasa por la notación binaria.

Por ejemplo, el número hexadecimal 3FB corresponde a:

3 = 1 1
F = 1 1 1 1
B = 1 0 1 1

es decir: 3FB = 1 1 1 1 1 1 1 0 1 1 = 1019 decimal

El mismo resultado se puede obtener multiplicando cada cifra hexadecimal por la correspondiente potencia de 16:

B = 11

$$3\ F\ B = 3 \times 16^2 + 15 \times 16^1 + 11 \times 16^0 = 3 \times 256 + 15 \times 16 + 11 = 1019$$

F = 15

10 /

2 cifras = 1 1 = decimal 3
4 cifras = 1 1 1 1 = decimal 15
8 cifras = 1 1 1 1 1 1 1 1 = decimal 255

TABLA 3

I	J	AND	OR	XOR	J+(NOT I)+1
11	65	1	75	74	54
11	66	2	75	73	55
11	67	3	75	72	56
11	68	0	79	79	57
11	69	1	79	78	58
11	70	2	79	77	59
11	71	3	79	76	60
11	72	8	75	67	61
11	73	9	75	66	62
11	74	10	75	65	63
11	75	11	75	64	64
11	76	8	79	71	65
11	77	9	79	70	66
11	78	10	79	69	67
11	79	11	79	68	68
11	80	0	91	91	69
11	81	1	91	90	70
11	82	2	91	89	71
11	83	3	91	88	72
11	84	0	95	95	73
11	85	1	95	94	74
11	86	2	95	93	75
11	87	3	95	92	76
11	88	8	91	83	77
11	89	9	91	82	78
11	90	10	91	81	79

TABLA 4

I	J	AND	OR	XOR	J+(NOT I)+1
32	65	0	97	97	33
32	66	0	98	98	34
32	67	0	99	99	35
32	68	0	100	100	36
32	69	0	101	101	37
32	70	0	102	102	38
32	71	0	103	103	39
32	72	0	104	104	40
32	73	0	105	105	41
32	74	0	106	106	42
32	75	0	107	107	43
32	76	0	108	108	44
32	77	0	109	109	45
32	78	0	110	110	46
32	79	0	111	111	47
32	80	0	112	112	48
32	81	0	113	113	49
32	82	0	114	114	50
32	83	0	115	115	51
32	84	0	116	116	52
32	85	0	117	117	53
32	86	0	118	118	54
32	87	0	119	119	55
32	88	0	120	120	56
32	89	0	121	121	57
32	90	0	122	122	58

En las tablas de la pág. 78 figuran los resultados de aplicar los operadores AND, OR, XOR y el operador compuesto J + (NOT I) + 1 a los números I = II y J de 65 a 90 (tabla 3), e I = 32, J de 65 a 90 (tabla 4).
La serie de números del 65 al 90 puede tener, en un ordenador, un significado muy especial.
En las tablas 3 y 4, la columna J + (NOT I) + 1 es la diferencia entre los números J e I.
En efecto, la expresión J + (NOT I) + 1 es la diferencia entre J e I calculada con el método del complemento (NOT I es el complemento de I).

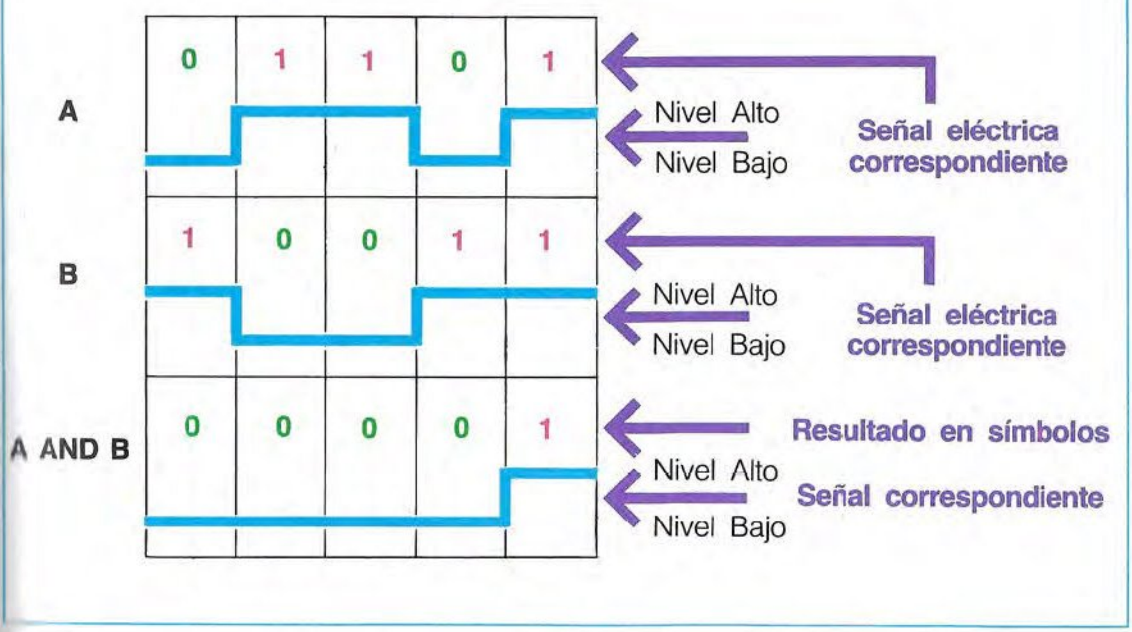
Lógica cableada

Los operadores lógicos pueden aplicarse a las señales eléctricas de la misma manera en que se aplican a los símbolos **1** y **0**. Las señales eléctricas digitales son una representación física de los símbolos **1** y **0**.
Mediante la consabida analogía:

señal eléctrica Alta = 1

señal eléctrica Baja = 0

Los valores binarios A = **0 1 1 0 1**, B = **1 0 0 1 1** se representan con señales de nivel Alto por cada símbolo **1** y señales de nivel Bajo por cada símbolo **0** (ver gráfico inferior).
La operación necesaria para obtener la señal A AND B (elaborando las señales A y B) se realiza mediante un tipo especial de circuitos digitales: los **circuitos lógicos**. Para cada operador lógico existe el circuito correspondiente.
Cada uno de estos circuitos tiene una función bien definida; por lo tanto, para montar un aparato que realizara una función lógica, tendríamos que construir un circuito muy concreto.
El aparato en cuestión sería capaz de realizar la función deseada, pero sólo ésa.
Si tuviéramos que cambiar de función, habría que cambiar también el circuito que la realiza.
La construcción de un aparato de este tipo consiste en conectar entre sí eléctricamente cierto número de componentes (circuitos integrados); la operación de conectar eléctricamente varios componentes se denomina **cableado**, por lo que un circuito tal es de **lógica cableada**.
El nombre deriva del hecho de que las operaciones lógicas se consiguen mediante conexiones (cableado) entre los componentes.
El principal inconveniente de este tipo de lógica es obvio: para cada función a realizar se necesita un circuito diferente.
Por el contrario, el ordenador es un aparato de **lógica programable**, es decir, posee circuitos capaces de realizar diversas tareas, en función de las órdenes **(instrucciones)** dadas por el **programador**.
Sin embargo, también en los ordenadores muchas funciones accesorias son realizadas por lógica cableada.



Circuitos lógicos

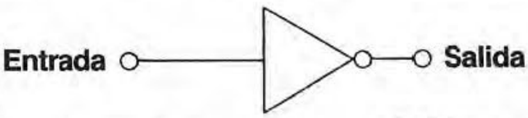
Como hemos visto, los principales operadores lógicos son:

- **NOT**
- **AND**
- **OR**
- **XOR**

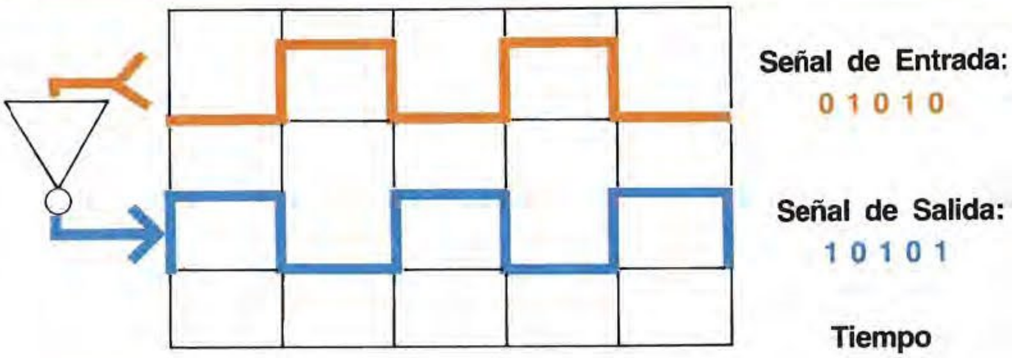
Estas funciones pueden ser realizadas por los correspondientes circuitos.
Para cada operador existe un circuito electrónico diferente capaz de realizar sus funciones.

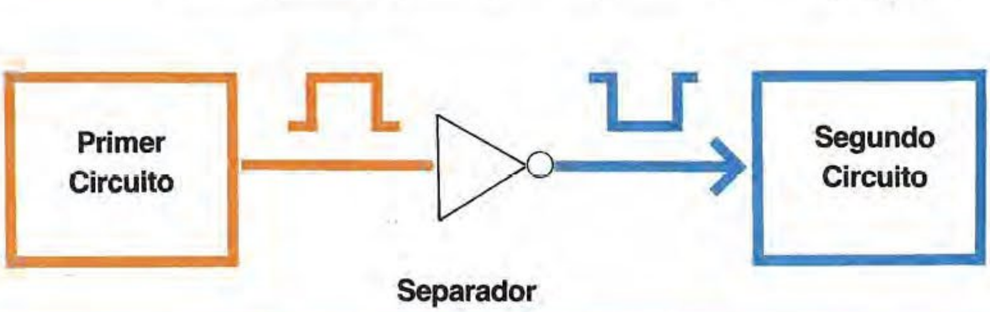
Operador NOT

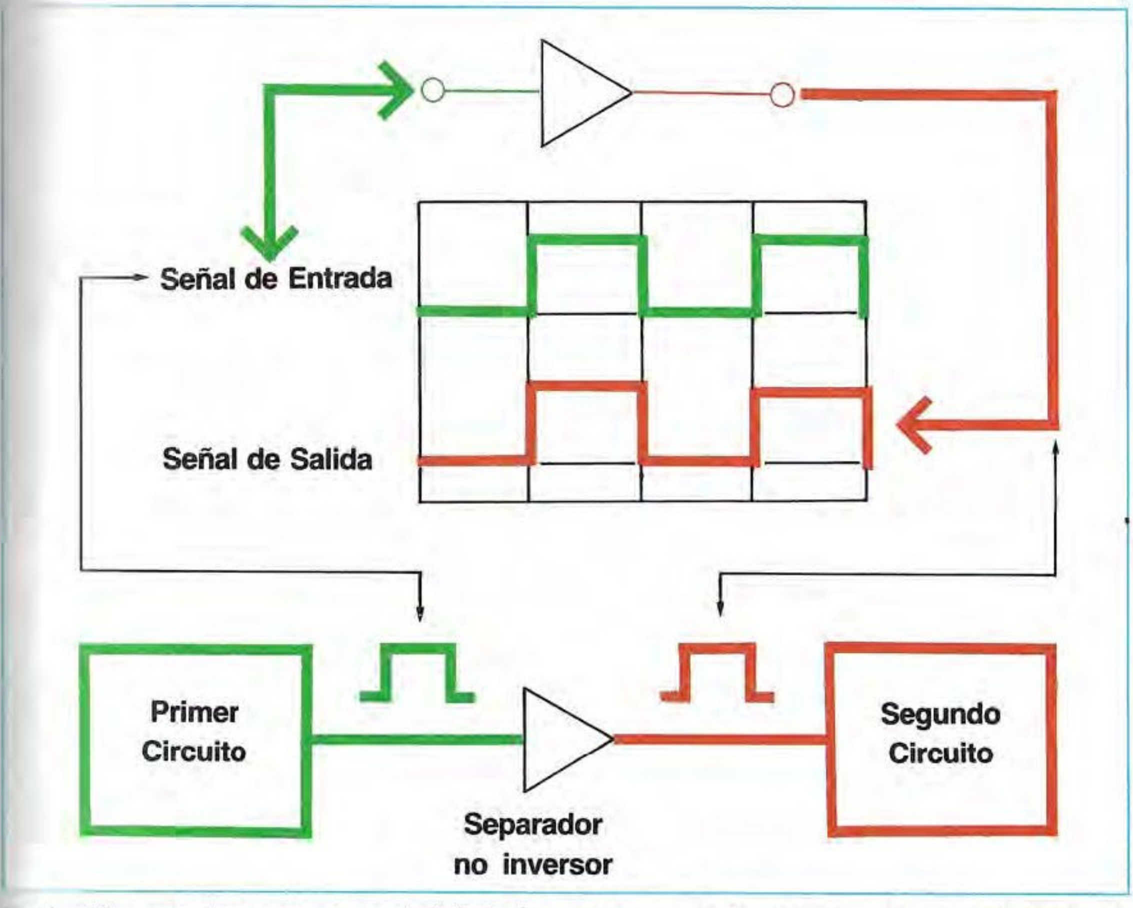
El operador NOT produce la inversión de un dato: el valor **1** se convierte en **0** y viceversa. Por lo tanto, el circuito capaz de realizar la función NOT no es más que un circuito cuya salida vale **0** (o **L**) cuando la entrada es **1 (H)**.
Si la entrada varía en el tiempo entre los estados **1 (H)** y **O (L)**, la salida varía de forma opuesta, es decir, entre **O (L)** y **1 (H)**. En el gráfico siguiente se muestra el símbolo de este circuito:



Si en la entrada insertáramos un CLOCK (ver «Los impulsos», págs. 45-47), la salida sería el opuesto de la señal del CLOCK de entrada.
Este tipo de circuito se denomina «de una entrada», puesto que tiene sólo una.
La pauta de las señales de entrada y salida se representa en el primer gráfico de abajo.
La función de este circuito no es sólo la de invertir la señal, sino también la de oficiar como separador entre otros tipos de circuito. Aclaremos esta función: en los ordenadores, las partes que constituyen el sistema realizan funciones complejas, con circuitos muy sofisticados que incluyen millares de componentes en pocos centímetros cuadrados. Esta alta densidad de componentes sólo se puede obtener mediante determinadas tecnologías de construcción.
Los circuitos construidos con estas tecnologías, precisamente por su estructura, no admiten mucha «potencia» y, por lo tanto, si se tiene que activar otro circuito que requiera potencia, hay que intercalar un tercer componente capaz de «aislar» entre sí los otros dos.
Véase el esquema lógico en el segundo gráfico de abajo, donde el separador es también un inversor, puesto que estos circuitos dan, en la salida, la señal de entrada invertida.
En algunas aplicaciones esta inversión puede resultar molesta; otros circuitos realizan las mismas funciones de separación, pero sin invertir la señal.







En el gráfico superior se muestran el símbolo, la pauta de señales y un esquema lógico de aplicación del circuito en cuestión.

Este tipo de circuito tiene también otra importante aplicación: su uso como amplificador.

Un amplificador es un circuito capaz de suministrar, en la salida, una señal proporcional a la de entrada, pero de valor más alto.

Hay amplificadores:

- de señal: cuando se amplifica el valor de la señal,
- de potencia: cuando se amplifica la potencia.

Veamos un ejemplo.

Supongamos que tenemos un ordenador que dirige una impresora situada en un local lejano. Para enviar los datos a la impresora se necesita una conexión mediante cable. Debido a la resistencia del cable y a las perturbaciones eléctricas procedentes del exterior, las señales llegarán a la impresora distorsionadas y debilitadas. Hará falta, por tanto, un amplificador instalado inmediatamente antes de la impresora, para «regenerar» el nivel y la forma de las señales.

Por otra parte, los circuitos del ordenador no pueden conectarse directamente a una línea de transmisión (cuando menos por motivos de seguridad): podría haber un cortocircuito en la línea, y por tanto hay que instalar un segundo amplificador entre el ordenador y la línea.

El gráfico inferior de la pág. 82 ilustra el sistema que acabamos de describir y las formas de onda correspondientes.

Para las aplicaciones de este tipo (transmisión de datos mediante línea) existen amplificadores (Line Drivers) de características especiales.

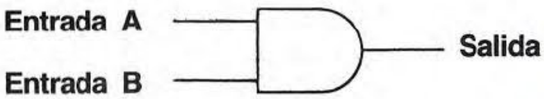
En realidad, las técnicas de reconstrucción de las señales son mucho más complejas de lo que se muestra en el gráfico.

Por último, señalemos que los circuitos de este tipo (denominados también **Buffer**) se utilizan tanto dentro de los ordenadores, para aislar entre sí los diversos componentes de la máquina, como fuera de ellos, para realizar las conexiones entre el ordenador y el mundo exterior.

Operador AND

En este caso el circuito es de dos entradas,

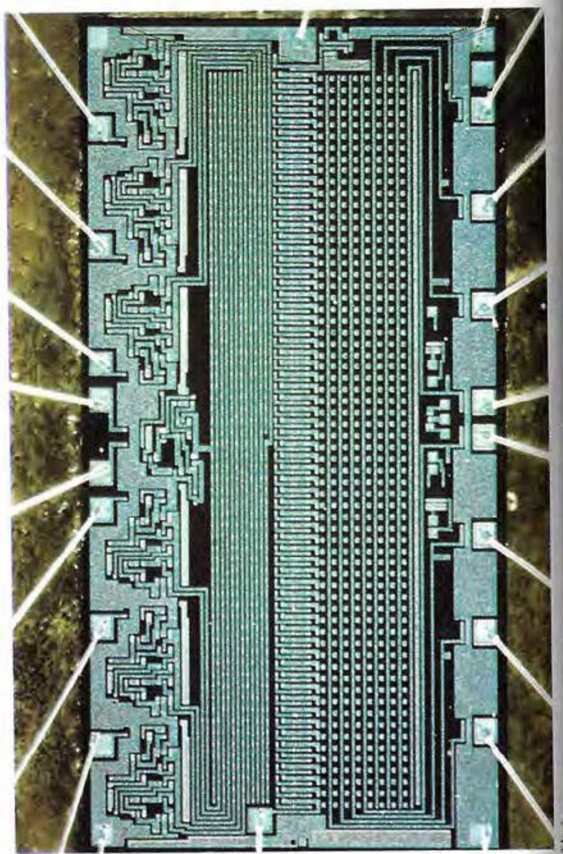
puesto que el operador AND precisa dos operandos (A AND B).
El símbolo gráfico del circuito es el siguiente:



Su funcionamiento es idéntico al del operador AND, recordando que el símbolo **1** se representa con la señal de nivel alto = **H**. Por lo tanto, la **TABLA DE LA VERDAD** es:

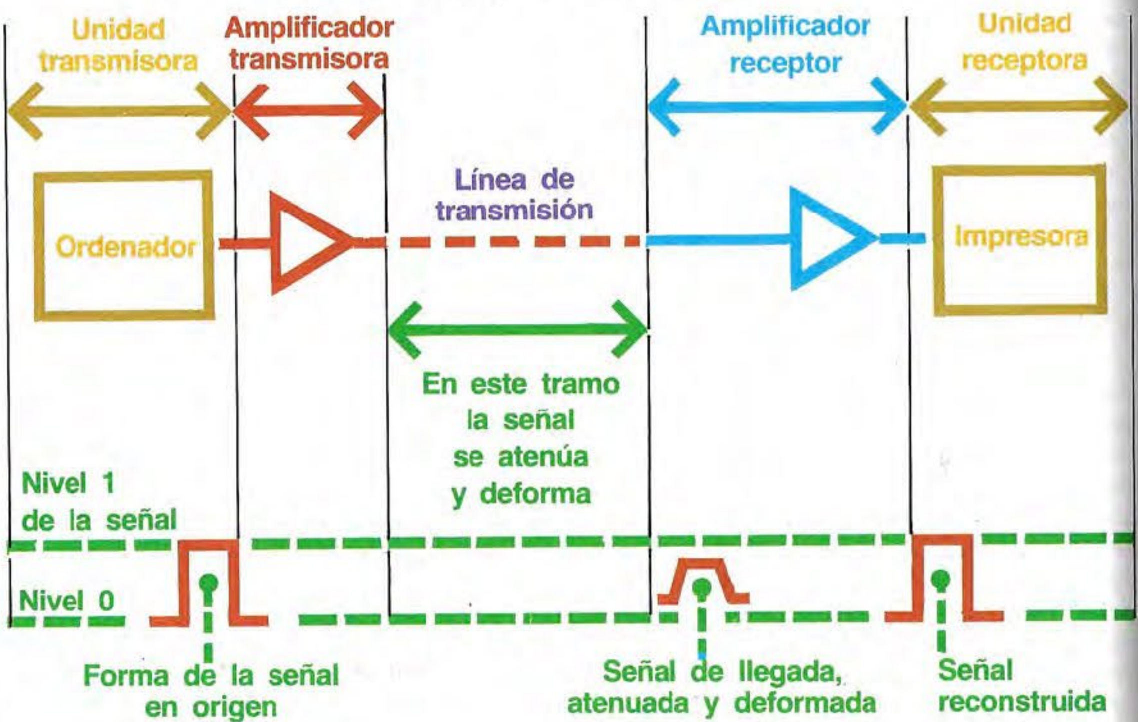
A	B	Salida = A AND B
H	**H**	**H**
H	**L**	**L**
L	**H**	**L**
L	**L**	**L**

y se obtiene directamente de la del operador AND, sustituyendo **1** por **H** y **O** por **L**.
Naturalmente, nada varía si se invierte la simbología, es decir, si se sustituye **1** por **L** y **O** por **H**. En este caso tendríamos un circuito con la lógica invertida, pero el funcionamiento no cambia.



Vista interior de un circuito integrado.

ESQUEMA DE TRANSMISION DE SEÑALES

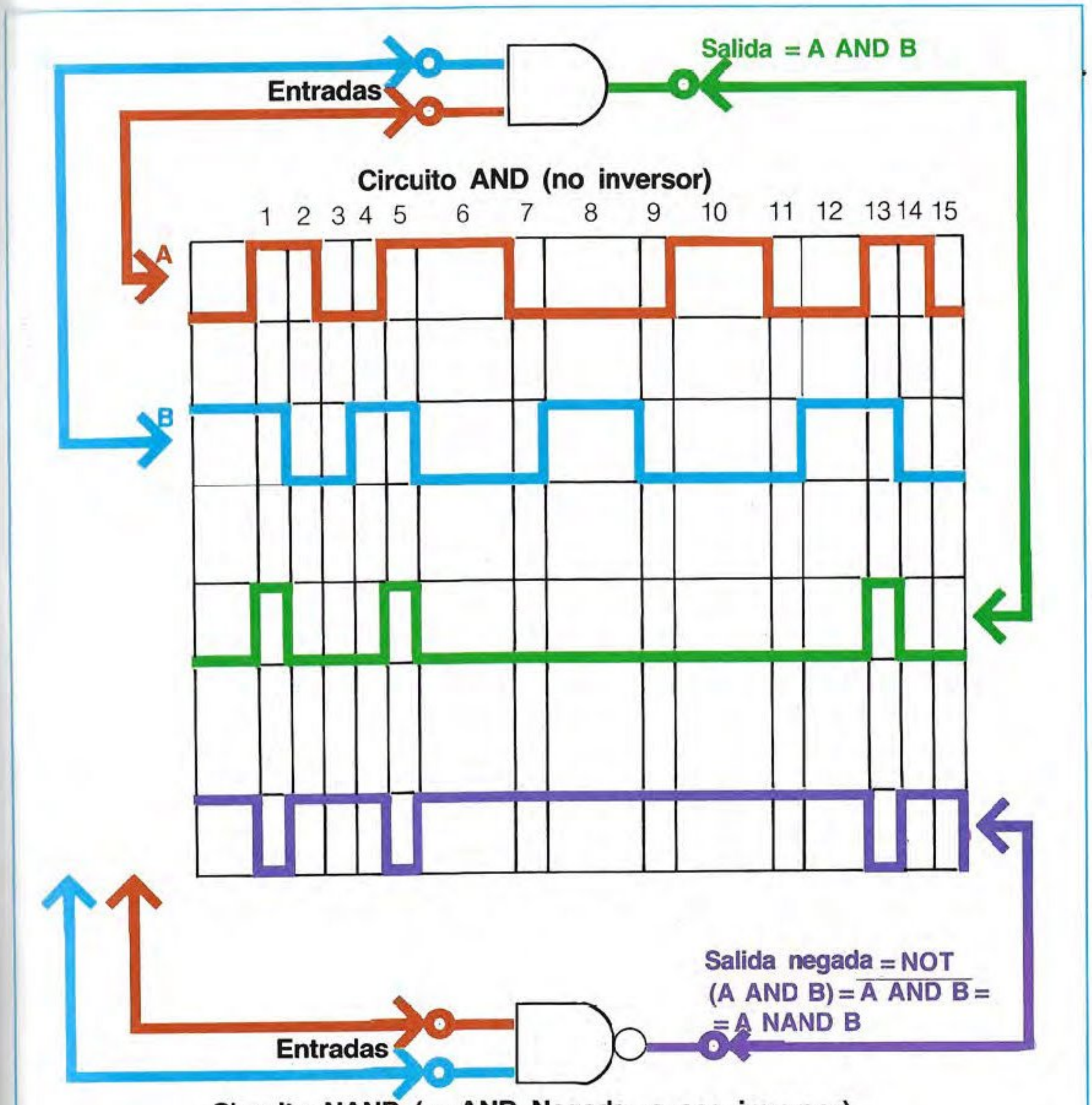


En el gráfico inferior se representan las formas de las señales.
En el mismo esquema se muestra también la versión inversora, denominada **NAND** (Negación de AND).
Para el circuito NAND, la **TABLA DE LA VERDAD** es:

A	B	NAND
H	H	L
H	L	H
L	H	H
L	L	H

que es la opuesta de la del circuito AND. O sea: si una de las dos entradas, por ejemplo la entrada A, se bloquea en el estado **H,** el circuito NAND se convierte en un inversor: quitando de la **TABLA DE LA VERDAD** las situaciones corresdientes al estado A = L (que no existe, ya que A está bloqueada en **H**), tenemos:

A	B	NAND
H	H	L
H	L	H
—	—	—
—	—	—



Circuito NAND (= AND Negado, o sea inversor)

Cómo se pregunta a un ordenador

Se ha aludido varias veces, en los capítulos anteriores, a las aplicaciones de los ordenadores a las tareas de gestión, y ahora es el momento de entrar en detalles con respecto a este tipo de utilización; nos referiremos especialmente al empleo de los operadores lógicos en los programas de acceso a un conjunto de datos memorizados en un archivo.

Por regla general, los grandes archivos automatizados requieren sistemas más potentes y capaces que los de los ordenadores personales, pero no son escasas las aplicaciones de este tipo en las que éstos pueden ser suficientes.

En el trabajo profesional es frecuente hacer que el ordenador gestione los archivos de datos que, de otro modo, requerirían largas y tediosas operaciones de búsqueda, a repetir con frecuencia. Es típico, por ejemplo, el caso de que una bibliografía especializada sobre un tema determinado sea clasificada por un investigador en un archivo personal gestionado por ordenador, o bien encomendar al ordenador personal la agenda telefónica familiar. Es necesario, en estos casos, disponer de un programa capaz de introducir en el archivo los nuevos datos que se considera útil conservar, y un programa que permita el acceso más rápido y seguro posible a cualquier dato concreto que interese extraer del archivo. Con lo que hemos visto hasta ahora, todavía no estamos en condiciones de codificar tales programas, pero de todos modos podemos reflexionar sobre los principios de la lógica a seguir.

En la pág. 16, por ejemplo, se expone la lógica de selección que permite el acceso a un determinado número de teléfono (dato) contenido en la guía de abonados (archivo). La búsqueda se puede esquematizar como un proceso de selección repetido en tres niveles, en función de condiciones cada vez más restrictivas, que reducen progresivamente la amplitud del campo en el que hay que buscar el dato.

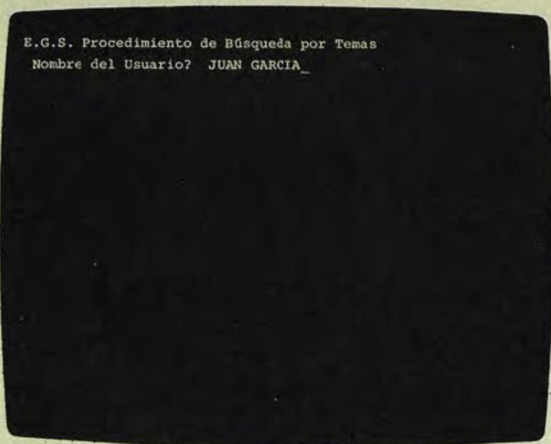
Podemos explicar más detalladamente la lógica de la búsqueda describiendo un procedimiento que permita obtener los tres niveles mencionados. Suponiendo que queramos buscar en el archivo el número de teléfono del señor Mario Pérez, que vive en la calle M. n.º 6, en el primer nivel tendremos que imponer la condición:

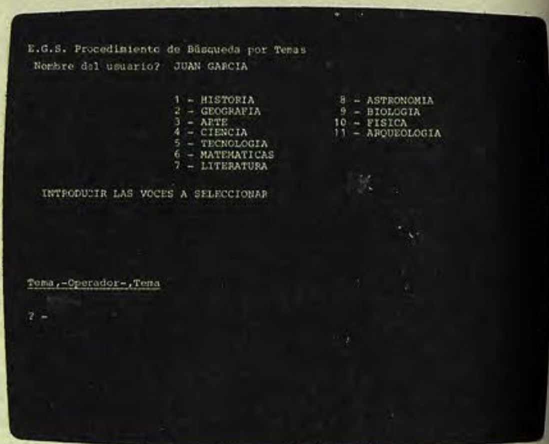
apellido = Pérez

Nuestro ordenador confrontará (de alguna de las maneras posibles, según como hayamos estructurado el archivo y el programa de búsqueda) los apellidos presentes en la memoria con el término de comparación (Pérez), y aislará el conjunto de todos los datos que cumplen la condición impuesta. Está claro que, si interrumpiéramos la búsqueda de este nivel, obtendríamos en la salida una notable masa de datos, ya que la condición impuesta no es muy selectiva; seguramente habrá en la guía muchos abonados que se apelliden Pérez. Es, pues, necesaria una selección de segundo nivel, que podemos obtener imponiendo la segunda condición:

nombre = Mario

Lo más importante es que las dos condiciones impuestas han de ser cumplidas simultánea-





mente por el dato que queremos extraer. Insertándolas por separado, sin conectarlas de forma lógica, tras imponer la primera tendríamos en la salida los números de todos los abonados que se apellidan Pérez, y tras imponer la segunda, los números de todos los abonados que se llaman Mario, independientemente de su apellido; ciertamente, no es eso lo que se desea. Las dos condiciones han de estar ligadas entre sí por el operador AND, de manera que formen la única condición global:

apellido = Pérez - AND - nombre = Mario

La otra condición restrictiva que tendremos que imponer en el tercer nivel es la relativa a la dirección del señor Pérez, que también deberá cumplirse junto con las demás. La condición global a suministrar al ordenador será, por tanto:

apellido = Pérez
(primer nivel)
-AND-
nombre = Mario
(segundo nivel)
-AND-
dirección = calle M. n.° 6
(tercer nivel)

La máquina nos suministrará entonces el número que buscamos, y podremos telefonear con toda seguridad al señor Pérez (a no ser que haya dos señores que se llamen Mario Pérez y vivan en la misma dirección; pero esta posibilidad también se puede prever).

Otra posibilidad es la de que, aunque recordemos el apellido de la persona que buscamos, no estemos seguros de si su nombre es Mario o Francisco. Sin embargo, la búsqueda puede efectuarse igualmente; los términos de comparación serán, esta vez, los siguientes:

1) apellido = Pérez
2) nombre = Mario o Francisco
3) dirección = calle M. n.° 6

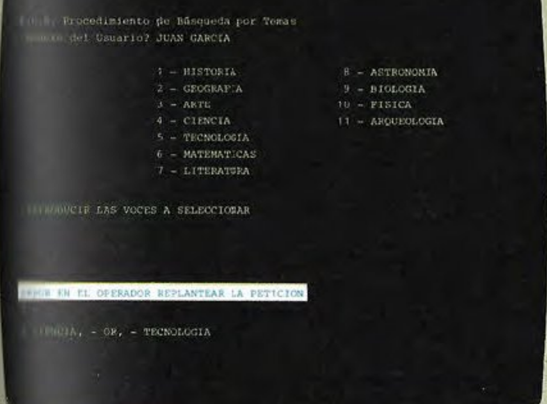
y las condiciones expuestas deberán también cumplirse simultáneamente. La condicion 2) se expresa, en términos operativos, de la forma que ya conocemos:

nombre = Mario-OR-Francisco

(uno u otro), y la búsqueda se efectuará en base a la condición global:

apellido = Pérez-AND-nombre = Mario OR-Francisco-AND-dirección = calle M. n.° 6

Nuestra incertidumbre sobre el nombre no perjudica, en este caso, la posibilidad de obtener con certeza el dato que buscamos, desde el momento en que existe la condición 3), que sirve de verificación de la condición 2). En su repaso de la zona del archivo que contiene los apellidos Pérez, la máquina se detendrá en cuanto encuentre un Mario o un Francisco, y comprobará que el dato correspondiente sea el que se busca leyendo la dirección y comparándola con la suministrada desde el exterior. En otras palabras, se dice que en este caso, como en el descrito anteriormente, hemos buscado un dato determinado de manera unívoca por la condición impuesta. Distinto habría sido si, además del nombre, hubiéramos olvidado tambié la dirección. El ordenador nos habría suministra-







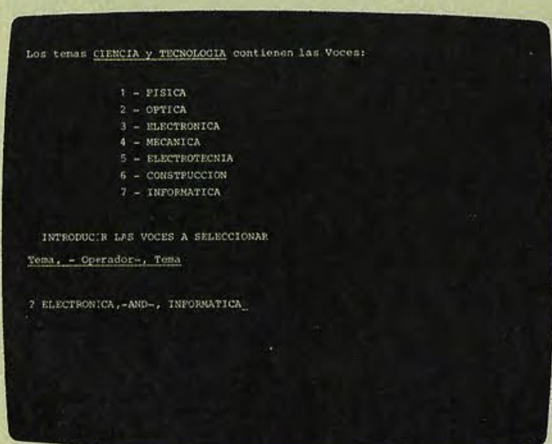
En una biblioteca, el ordenador agiliza el trabajo de archivo y permite una búsqueda más rápida.

do, en la salida, los números de teléfono de todos los Pérez incluidos en la guía, y hubiésemos tenido que pasarnos horas delante del monitor o de un largo listado. En tal caso, hubiéramos sufrido las molestias inherentes al hecho de buscar un dato imponiendo una condición que no lo determina de manera unívoca.

Un amplio campo de aplicación de este tipo de programas lo encontramos en las grandes bibliotecas, en las que cada texto disponible se clasifica en un record *que incluye el nombre del autor, el título, el tema (o temas) tratados, el idioma en que está escrito, la situación en los estantes, la disponibilidad actual, etc. La gestión automatizada de estos datos permite evitar el largo y tedioso trabajo de búsqueda (con el consiguiente riesgo de despistes y olvidos) que tenían que realizar los bibliotecarios de hace veinte años.*

En aquella época, las mayores dificultades se presentaban cuando había que localizar todos los textos que trataban un tema determinado.

Gracias a la creación de archivos automatizados y a la utilización de los operadores lógicos en los programas de búsqueda, hoy día esta tarea ha sido simplificada enormemente. Los





operadores lógicos permiten «organizar» los datos almacenados en la memoria de la manera más adecuada para suministrar la información que buscamos.

En las ilustraciones se muestra, a título de ejemplo, la utilización de un ordenador para localizar todos los textos disponibles en una biblioteca sobre el tema «memoria de ordenadores». El bibliotecario moderno pide desde su consola la introducción en memoria del programa de búsqueda a utilizar. Como respuesta, el ordenador pide el nombre del usuario. Este requisito puede tener diversos objetivos; puede servir, por ejemplo, para impedir que personas no autorizadas puedan utilizar el programa, o bien para limitar el acceso a determinados textos de índole reservada, o bien para poder diferenciar las peticiones efectuadas por los distintos usuarios.

La máquina presenta entonces en la pantalla un típico MENU con las principales ramas del saber, con las que habrá que operar de forma lógica. En el caso planteado, las secciones que pueden interesar son **Ciencia** y **Tecnología** y, por tanto, habrá que imponer la condición:

Ciencia, -OR-, Tecnología

La forma en que la instrucción ha de ser modulada en el teclado es sugerida por el ordenador mismo, que muestra en la pantalla la frase:

Tema, -Operador-, Tema

Es imprescindible respetar la sintaxis correcta para que el ordenador pueda reconocer la instrucción. Un eventual error en el formato provocaría la anulación de la instrucción y la aparición de un mensaje de advertencia en la pantalla; entonces habría que volver a insertar la instrucción con el formato correcto.

Una vez recibida la primera condición selectiva, la máquina responde suministrando el número global de textos clasificados en las secciones especificadas y, si este número no es demasiado elevado, el usuario puede pedir la impresión del listado con los datos de dichos textos. En nuestro caso, el número es demasiado elevado, y el listado no nos serviría de gran ayuda. Tenemos, por tanto, que seguir la búsqueda, y comunicaremos a la máquina esta intención mediante la instrucción pertinente.

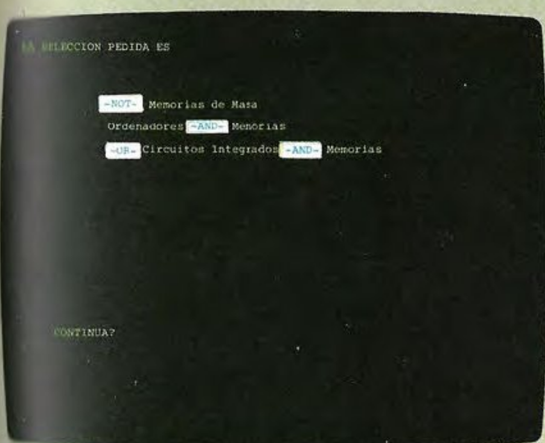
En la fase siguiente, el ordenador presenta en la pantalla todas las subsecciones incluidas en las secciones Ciencia / Tecnología. Lo que nos interesa podría encontrarse en las subsecciones Electrónica o Informática y, por tanto, tendremos que plantear la condición:

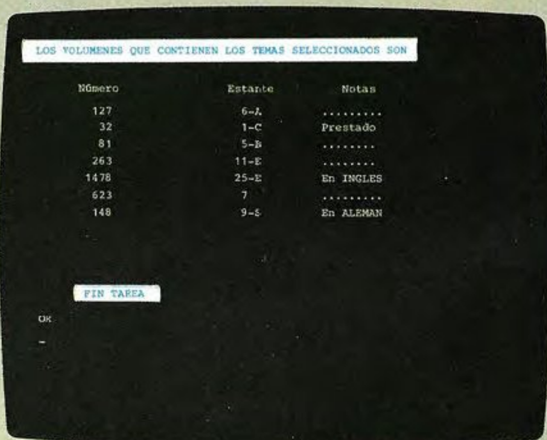
Electrónica, -AND-, Informática

Esta selección implica una notable reducción del número de textos, y en este momento podemos efectuar la selección definitiva, insertando la condición:

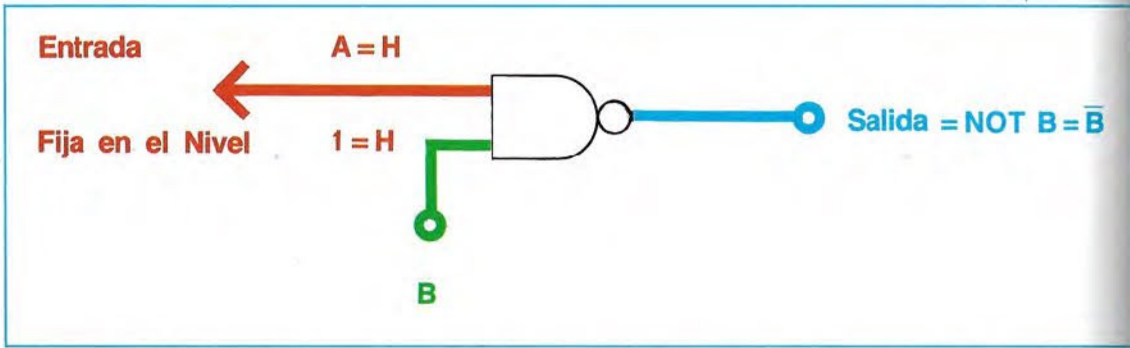
Circuitos integrados, -AND-, Memorias

Si hubiéramos planteado sólo la condición Memorias, habríamos obtenido en la salida todos los textos relativos a las memorias y, por tanto, también sobre memorias de masa (cintas, discos); la especificación Circuitos integrados nos permite, sin embargo, aislar sólo los textos que tratan de memorias a base de circuitos integrados. El ordenador puede ahora suministrarnos el listado con los datos de los textos seleccionados.





La salida es el opuesto de la entrada B, es decir, la salida es la negación de B (NOT B). Simbólicamente, si la entrada es B, la salida suministrada por este circuito, es decir, el opuesto de B (negación de B), se indica en la forma $\overline{B}$. El esquema aparece en el gráfico siguiente:



Vemos aquí representadas dos entradas genéricas (A y B) con formas de onda no periódicas. Para deducir la salida, obsérvese que la señal de nivel alto equivale a **1**, mientras la señal de nivel bajo equivale a **0**.
De este modo, como demuestra el gráfico de la pág. 83, tenemos:

Tramo 1:
los niveles de A y B son altos, es decir: A = **1**, B = **1**, y por tanto A AND B = **1**, o sea alto: **H**.

Tramo 2:
A está alto, es decir, A = **1**; B está bajo, o sea B = **0**, y por tanto tenemos:
A AND B = **0**, o sea bajo: **L**.

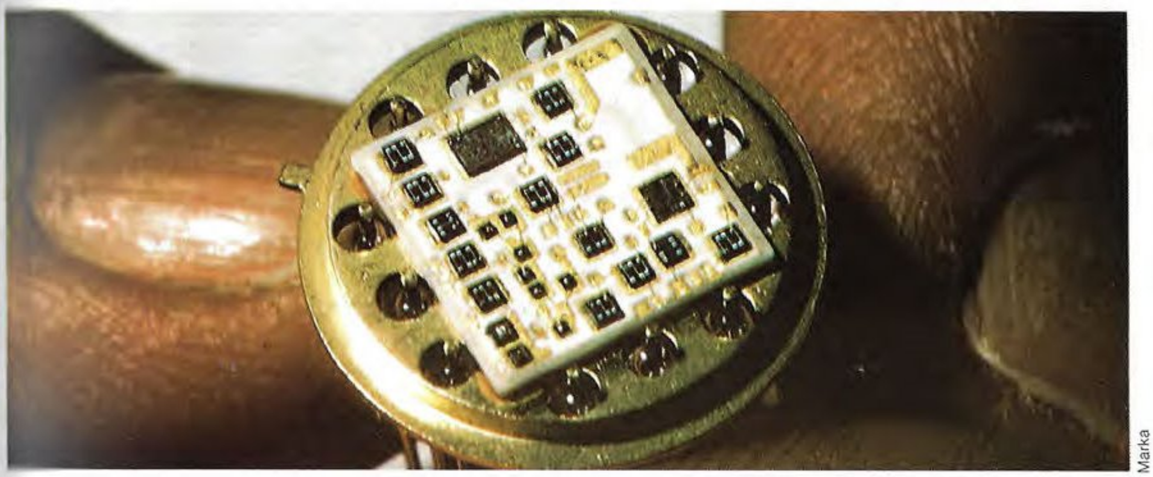
En la tabla adjunta se especifican los 15 tramos del gráfico de la pág. 83.

Tramo	A	B	A AND B	$\overline{A \text{ AND } B}$
1	**1**	**1**	**1**	**0**
2	**1**	**0**	**0**	**1**
3	**0**	**0**	**0**	**1**
4	**0**	**1**	**0**	**1**
5	**1**	**1**	**1**	**0**
6	**1**	**0**	**0**	**1**
7	**0**	**0**	**0**	**1**
8	**0**	**1**	**0**	**1**
9	**0**	**0**	**0**	**1**
10	**1**	**0**	**0**	**1**
11	**0**	**0**	**0**	**1**
12	**0**	**1**	**0**	**1**
13	**1**	**1**	**1**	**0**
14	**1**	**0**	**0**	**1**
15	**0**	**0**	**0**	**1**

Circuitos integrados ensamblados en una placa.



Marka



Un chip de silicio y zafiros, muy aumentado.

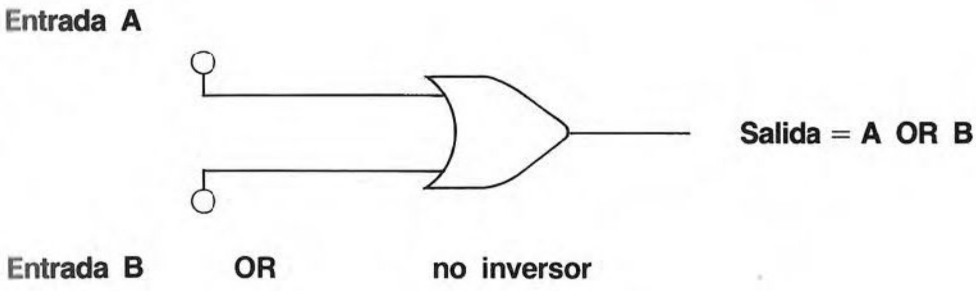
Operador OR

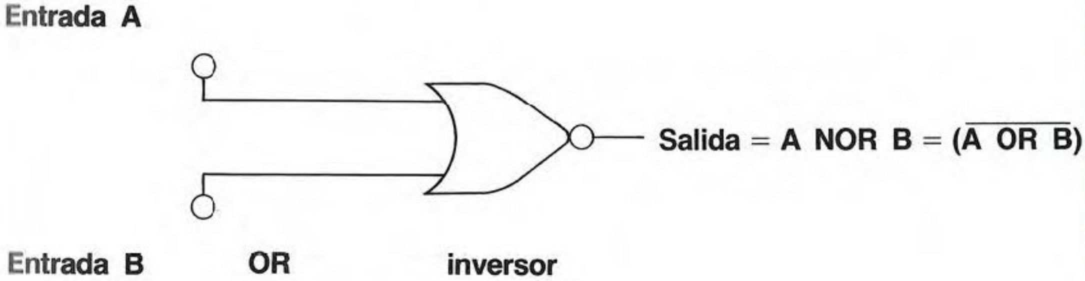
Con la consabida notación: 1 = **H** y 0 = **L**, el circuito que realiza la función OR tiene la misma tabla de la verdad que el operador homónimo. También para estos circuitos existen las dos versiones: no inversora e inversora. Los símbolos se muestran en el gráfico inferior.

La tabla del OR inversor se obtiene invirtiendo la del OR no inversor:

A	B	OR	$\overline{OR}$ *
H	H	H	L
H	L	H	L
L	H	H	L
L	L	L	H

* Un trazo horizontal sobre el símbolo de un operador lógico significa el mismo operador invertido, es decir, NEGADO. Por tanto, el símbolo $\overline{OR}$ significa OR negado.





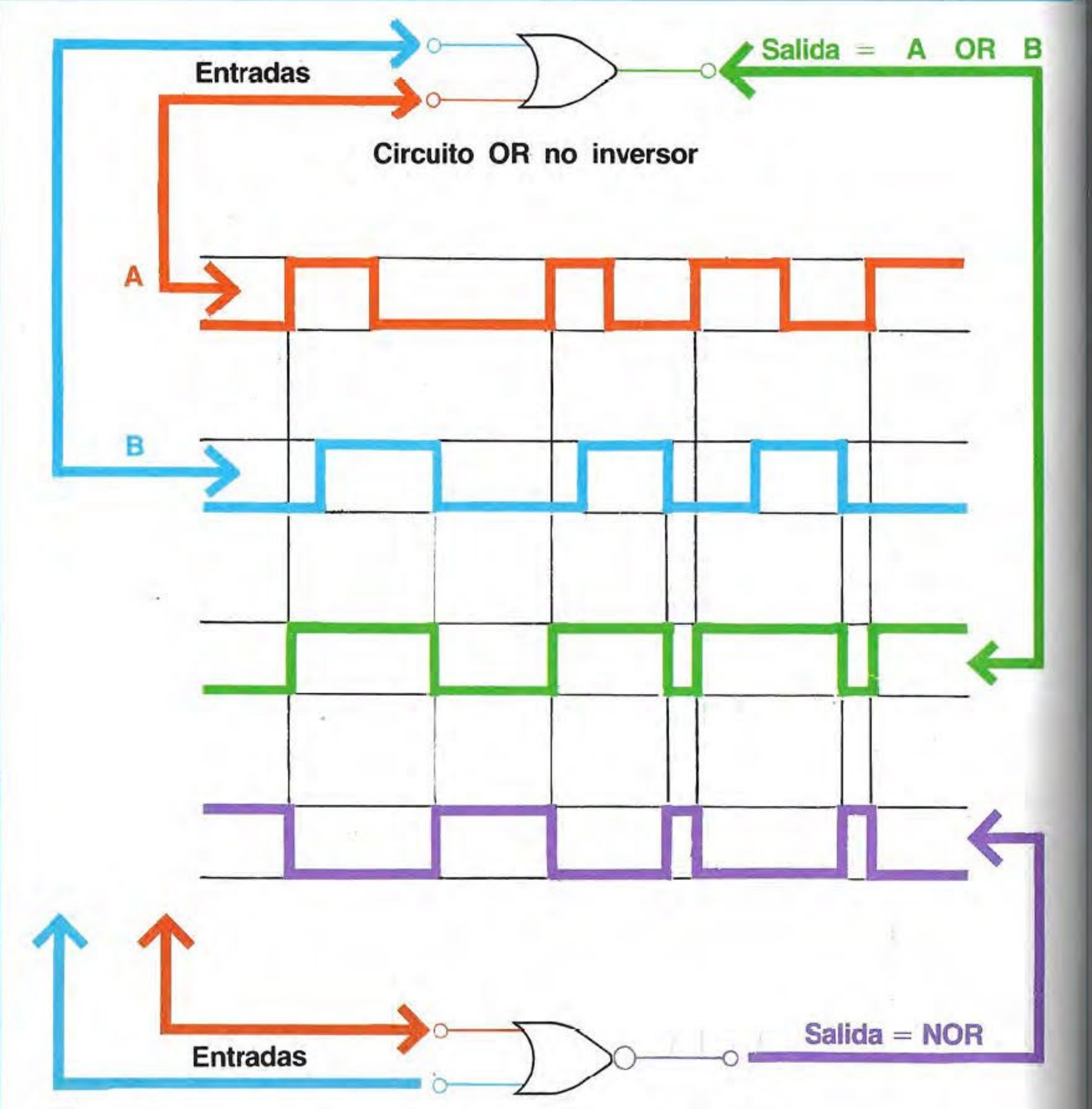
En el gráfico inferior se muestra un ejemplo de formas de onda.
También en este caso se supone que en ambas entradas hay una señal no repetitiva (no periódica).

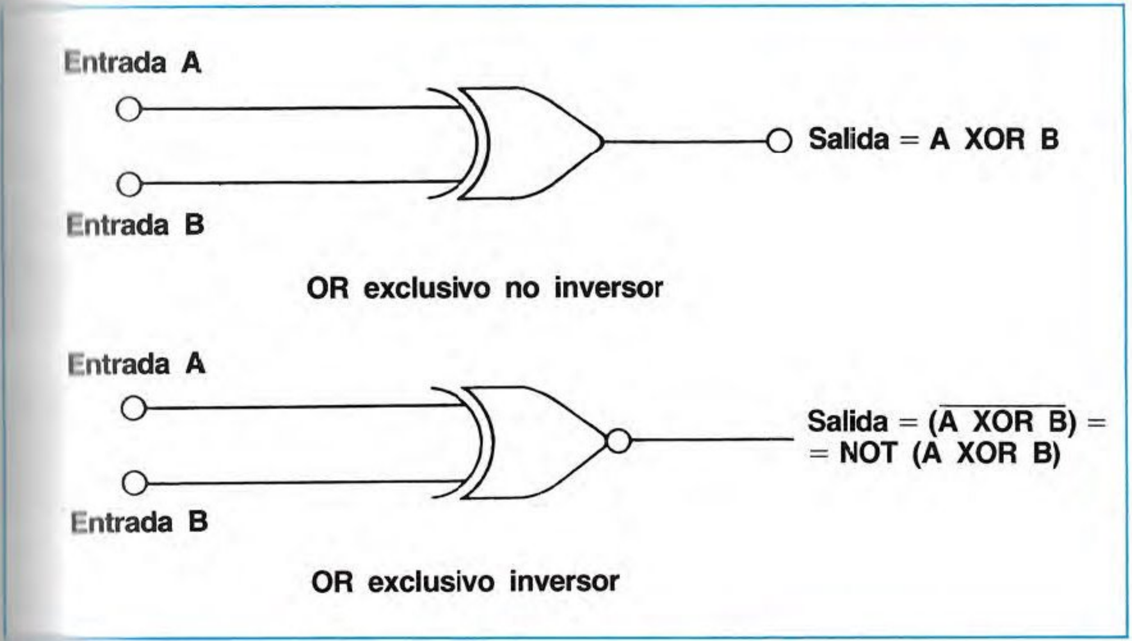
Operador XOR

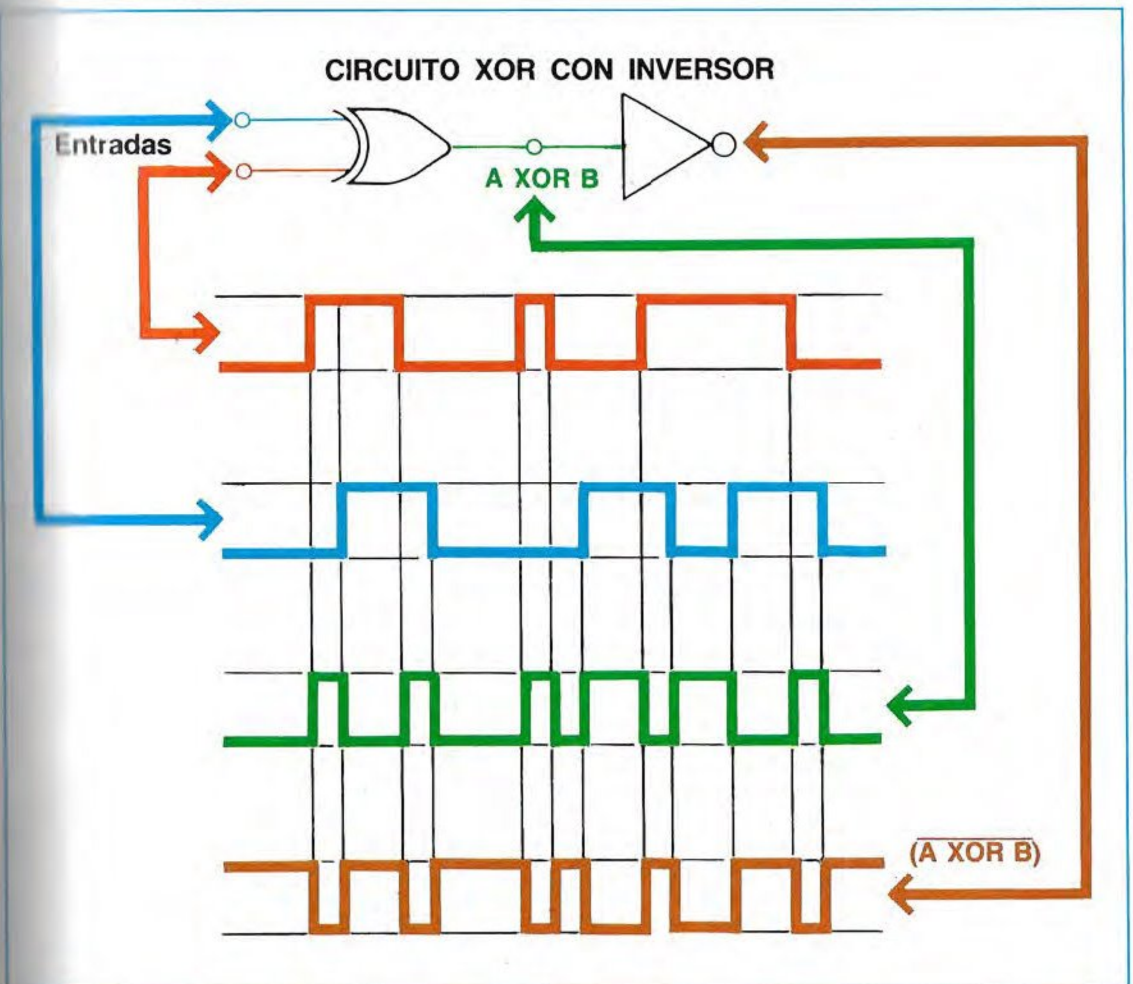
También para este circuito se aplica la **TABLA DE LA VERDAD** del operador homónimo. Los símbolos se muestran en el gráfico superior de la pág. 91.
Para este circuito es muy difícil encontrar fabricantes que faciliten el tipo inversor ($\overline{A \text{ XOR } B}$).
Pero dicha función puede realizarse acoplando un inversor a continuación del circuito OR no inversor.
En el gráfico inferior de la pág. 91 se muestran la conexión y las formas de onda.

Otros tipos de inversor

En el circuito amplificador (inversor o no) hay una sola entrada.
Todos los demás circuitos tienen dos (o más); sin embargo, pueden homologarse al primero (excepto el circuito XOR) sin más que conectar juntas las dos entradas.




Entrada A
Salida = A XOR B
Entrada B
OR exclusivo no inversor
Entrada A
Salida = $(\overline{A \text{ XOR } B})$ =
= NOT (A XOR B)
Entrada B
OR exclusivo inversor


CIRCUITO XOR CON INVERSOR
Entradas
A XOR B
$(\overline{A \text{ XOR } B})$

El banco del futuro

En todo el mundo, los bancos de mediana y gran envergadura están trabajando para definir cuáles y cómo serán los servicios ofrecidos a sus clientes con la introducción de las nuevas tecnologías disponibles. Cualquiera que sea el resultado de estos desarrollos, el aspecto de las transacciones bancarias normales quedará irrevocablemente modificado.

Aunque aún no se puede definir claramente la índole de los nuevos servicios, su nivel de sofisticación será tal que, en comparación, las actuales «maquinitas» del tipo Automatic Teller Machines resultarán primitivas.

Los bancos están trabajando en diversos proyectos con tal celeridad y reserva, que las confederaciones industriales y comerciales empiezan a preocuparse seriamente. Incluso en el interior de cada banco concreto, las expectativas y ambiciones de las nuevas promociones de técnicos ponen en entredicho a los máximos cuadros directivos de la generación anterior.

Pero, al margen de estas consideraciones, es indudable que los nuevos servicios implicarán cada vez más a los clientes en la ejecución de buena parte de las transacciones, en lugares distintos de las agencias bancarias mismas. Los bancos están transformando sus servicios en self-service, *mediante dispositivos a los que los clientes tendrán acceso directo desde sus propias casas, oficinas o puntos de utilización del dinero. Ello no significa la desaparición de las ventanillas bancarias: se trata, en todo caso, de una redistribución de las actividades encaminada a automatizar y descentralizar hacia los puntos de origen las operaciones rutinarias, dejando a las agencias mayor margen para los contactos con clientes que planteen situaciones especiales o, en cualquier caso, no resolubles mediante simples transacciones.*

En muchos países, diversos bancos están llevando a cabo experimentos para comprobar «en vivo» los efectos y el funcionamiento de los servicios de nueva concepción.

El banco inglés más avanzado en el estudio de los sistemas «caseros», el Barclays Bank, podrá, dentro de muy poco tiempo, ofrecer a sus clientes un self-service a través de terminales personales instaladas en casas y oficinas.

Terminales del sistema informático del Chase Manhattan Bank.



Marka



1 En una entidad bancaria del otro lado del océano se efectúa una operación.
2 Los datos relativos a la operación son transmitidos a Europa por radio.
3 La sede europea de la entidad bancaria es informada de la operación efectuada en el otro continente.
4 En tiempo real, sin dilación alguna, el movimiento bancario realizado a miles de kilómetros es operativo en todas las sedes conectadas con la red.

■ Las estaciones «de cabeza» sirven de concentradores para los satélites. En dichas estaciones se montan las complejas instalaciones necesarias para el puente de radio.
• Estaciones satélite conectadas por cable con las estaciones principales (■).

Un servicio análogo será ofrecido por los dos mayores bancos neoyorquinos, el Citibank y el Chemical Bank, a unos doscientos clientes que forman parte de un muestreo. Los dos bancos suministrarán a estos clientes sendos terminales personales (caseros) mediante los que podrán realizar una serie de transacciones y consultas.

En Alemana Occidental, un pequeño banco, el Verbraucher Bank, ha iniciado hace poco un experimento con una muestra de su clientela: se ha suministrado a los clientes seleccionados un pequeño terminal conectado, mediante un servicio tipo Videotel, a un ordenador central, y a través de él podrán comprobar sus saldos, obtener un extracto de cuenta, efectuar transacciones a la propia cuenta o a otros bancos. Si bien el experimento ha sido iniciado por un banco menor, otros grandes bancos alemanes se han visto obligados a seguir su ejemplo.

Todavía más «futurista» es el experimento iniciado en enero de 1982 por los bancos franceses en dos poblaciones-muestra, en las que a todos los habitantes que tienen una cuenta corriente les ha sido suministrado un módulo elec-



Fortronic

Terminal bancario (1) con display (2); impresora (3); teclado (4); lector de fichas magnéticas (5).

trónico preprogramado y un terminal casero especial. Cada «tarjeta-memoria», del tamaño de una tarjeta de crédito normal, contiene un pequeño microprocesador, una memoria ROM con los programas y algoritmos especiales de control y una memoria escribible pero no cancelable capaz de almacenar un considerable número de transacciones. Esta tarjeta se utiliza, en sustitución del talonario de cheques, en las ventanillas bancarias y en los puntos de venta más dispares (supermercados, estaciones de servicio, cabinas telefónicas, lugares de recreo, etc.). Paralelamente, los puntos de venta han sido dotados con terminales especiales POS (terminales de caja) capaces de operar tanto con las tarjetas de crédito normales como con las tarjetas-memoria: en este último caso, la «inteligencia» utilizada para efectuar los controles no es la del terminal sino la de la tarjeta.

En Bélgica, la Caisse Générale d'Épargne et de Retraite está llevando a cabo un experimento relativo a la reorganización de las actividades de agencia, paralelamente a una reestructuración de los servicios de pago y cobro.

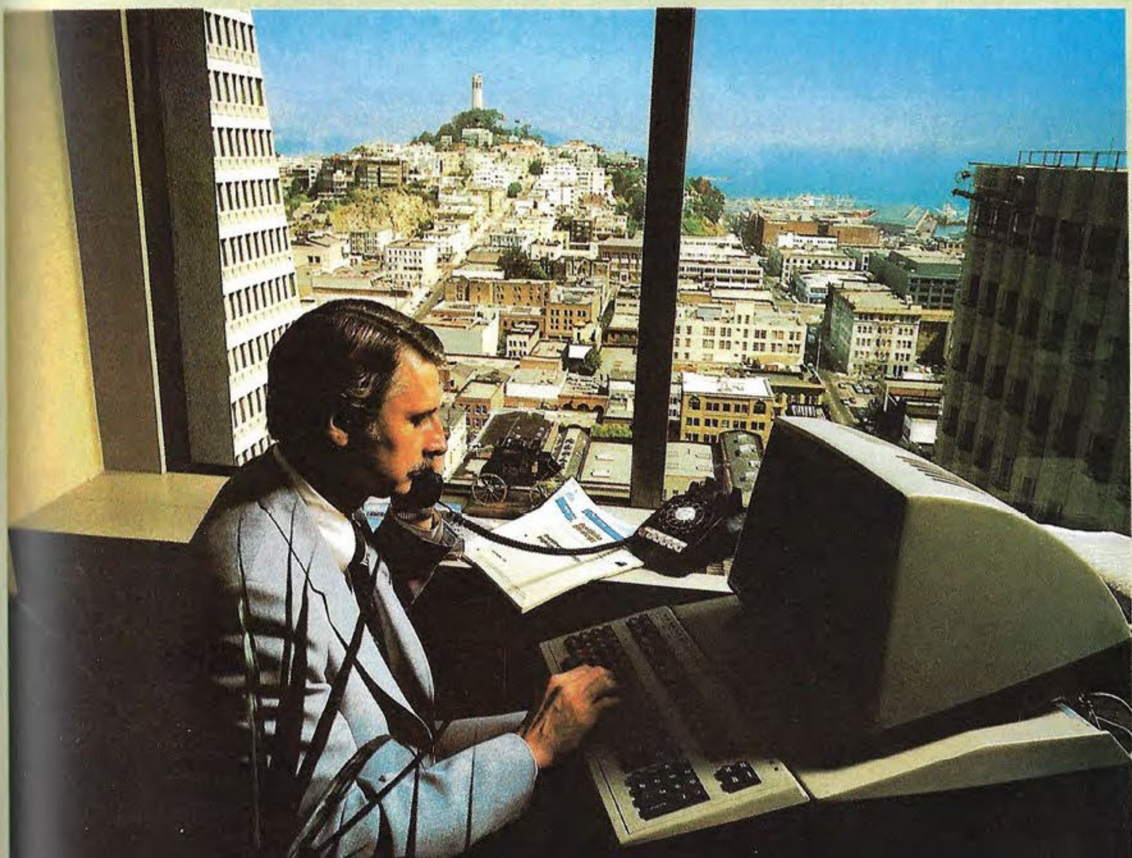
En las nuevas agencias, totalmente replanteadas para garantizar un contacto más personal con la clientela, unos paneles de consulta especiales, conectados con un ordenador, «guía» a los clientes a despachos abiertos al público o privados en los que tienen lugar los contactos con el personal del banco. La gestión de las colas de espera y el reparto de las tareas están, de este modo, totalmente automatizados, y permiten un desarrollo más rápido y humano de las transacciones y actividades que requieren un contacto directo con el cliente.

Self-service bancario significa sustituir al empleado por una pantalla de televisión en la que se puede visualizar la información y un teclado mediante el que enviar instrucciones a un ordenador del banco. El teclado y la pantalla, con el eventual complemento de una pequeña impresora, constituyen un terminal que puede ser instalado directamente en casa (en cuyo caso el

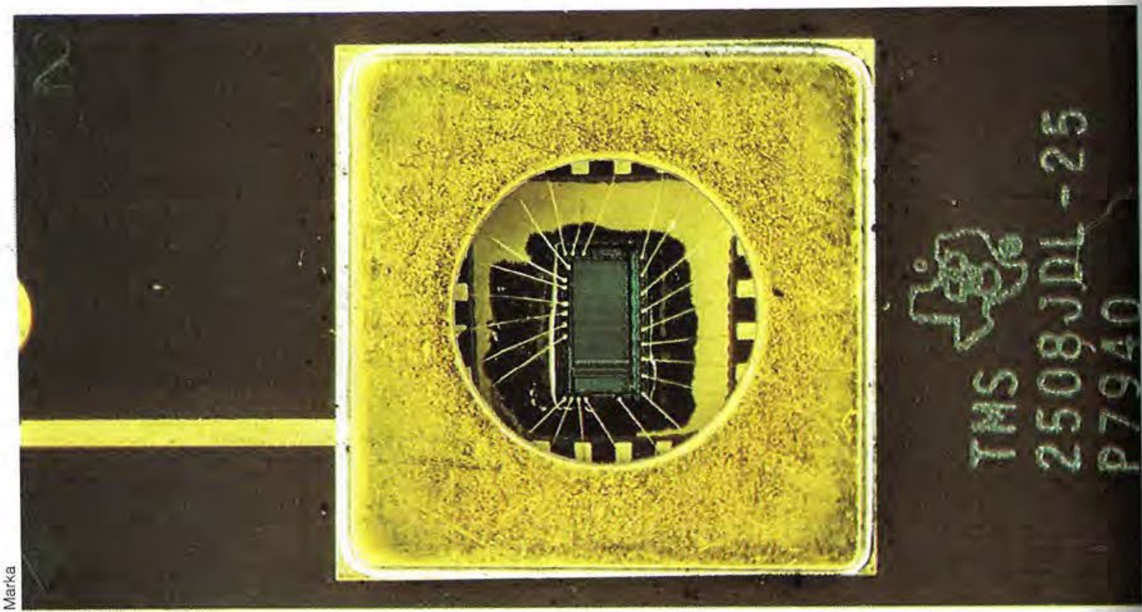
televisor normal hace las veces de pantalla), en la oficina, en la fábrica o en las agencias bancarias. En Japón hay terminales de este tipo distribuidas a lo largo de las paredes de las agencias, como si fueran teléfonos públicos.
La conexión entre el terminal casero y el ordenador del banco se efectúa, simplemente, mediante línea telefónica: el cliente llama al ordenador a través del terminal, marca su código de identificación y una palabra de instrucción, e inicia el diálogo guiado por una serie de listas orientativas. En lugar de teclear el código, éste puede ser leído directamente en la banda magnética de una tarjeta de crédito.
Muchos bancos estadounidenses han invertido directamente capital en el desarrollo de sus propios terminales; por ejemplo, el Chemical Bank y el Citibank. El costo actual varía entre las cincuenta y las ciento cincuenta mil pesetas aproximadamente, pero el futuro desarrollo masivo llevará el precio medio a unas 15.000 ptas.
En Europa, sin embargo, los bancos parecen inclinarse preferentemente por los sistemas del tipo Videotel, utilizando así un servicio ya disponible para los abonados telefónicos. Los diversos servicios nacionales de este tipo pronto estarán conectados entre sí, brindando la posibilidad de acceso a la propia cuenta corriente desde cualquier terminal Viewdata de Europa. Efectivamente, los bancos están desarrollando un software especial que permite el acceso a los propios archivos desde terminales de este tipo, mediante mandos y controles especiales.
Naturalmente, los dos proyectos, el estadounidense y el europeo, no son incompatibles, puesto que un terminal casero puede ser adaptado fácilmente para que sirva de terminal Viewdata. Un terminal casero es más potente que un televisor adaptado, y también podría utilizarse localmente como ordenador personal.
El Chemical Bank norteamericano, en el marco del experimento antes citado, no permite a sus clientes el acceso directo a los archivos del banco, sino sólo a una copia registrada en un ordenador dedicado; dicha copia es elaborada y actualizada al final de cada jornada laboral.
Por el contrario, el Verbraucher Bank alemán permite el acceso directo a los propios archivos, mediante un control muy estricto basado no sólo en el número de cuenta, sino más bien en tres palabras de instrucción sucesivas, que pueden ir cambiando periódicamente.

(Fragmento de un artículo de C. Merli y L. Gibin en la publicación italiana INFORMATICA 70, n.° 95, marzo 1982.)

La facilidad de uso de los ordenadores modernos los hace accesibles incluso para operadores escasamente familiarizados con la electrónica.



Prime



Marka

En un dispositivo con dos entradas, puesto que cada una puede hallarse en los estados **H** y **L**, las combinaciones posibles son:

Primera Entrada	Segunda Entrada
H	L
H	H
L	H
L	L

Si las dos entradas se conectan juntas, se hallarán siempre en el mismo estado: las dos en **H** o las dos en **L**.
Por lo tanto, los estados posibles son:

A	B
H	H
L	L

En este chip de silicio, auténtica joya de la electrónica, se ven claramente las conexiones entre el módulo y las «patitas».

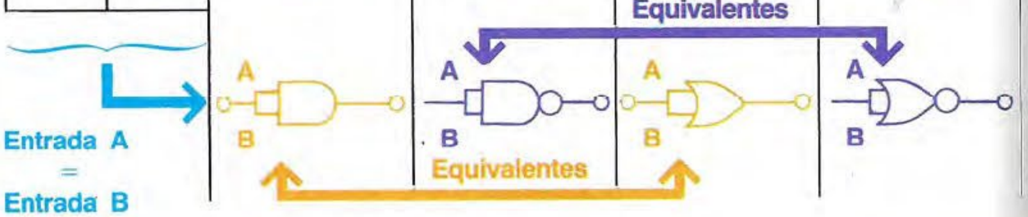
Aplicando a estos estados los operadores AND, NAND, OR y NOR, se obtiene el gráfico inferior. Como puede verse, con este tipo de conexión tenemos:

AND = OR = amplificador no inversor
NAND = NOR = amplificador inversor

El circuito XOR no puede conectarse de esta manera, puesto que entonces su salida siempre sería:

L (**H** XOR **H** = **L**; **L** XOR **L** = **L**).

A	B	A AND B	A NAND B	A OR B	A NOR B
H	H	H	L	H	L
L	L	L	H	L	H



Circuitos integrados

En los capítulos anteriores hemos hablado de circuitos capaces de realizar determinadas funciones y los hemos examinado desde el punto de vista lógico; sin embargo, no hemos visto cómo están hechos dichos circuitos.

En los comienzos de la electrónica digital, los circuitos estaban formados por componentes discretos, es decir, fabricados por separado y luego conectados entre sí para constituir el circuito. Posteriormente se consiguió englobar todos los elementos de un circuito en una única plaqueta, obteniendo así con un solo componente las mismas funciones para las que antes se necesitaban docenas de piezas separadas. Este nuevo tipo de componente se denominó CIRCUITO INTEGRADO.

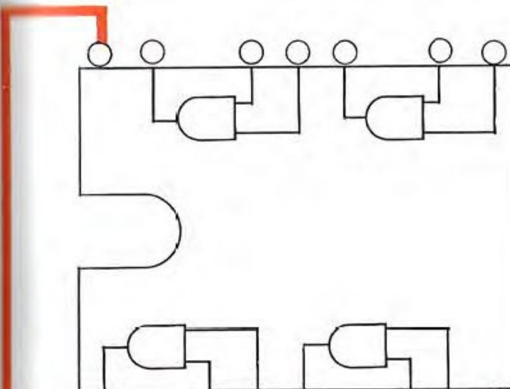
La utilización de circuitos integrados es muy simple, ya que, aparte de las conexiones de entrada y salida, no hay que ocuparse más que de la alimentación eléctrica.

El valor de la alimentación depende del tipo de circuito integrado: los más comunes (llamados circuitos TTL por su estructura física) sólo necesitan 5 voltios y pueden funcionar con pilas.

Normalmente, en un único bloque hay varios circuitos completos. Por ejemplo, uno de los más comunes es el integrado que realiza la función AND, en cuyo interior hay cuatro (o más) de estos circuitos.

Este integrado tiene 14 patillas, puesto que necesita 3 para cada circuito (ya que cada circuito AND tiene 2 entradas + 1 salida) y 2 más para la alimentación eléctrica.

Los fabricantes de estos integrados suministran, para cada tipo, el esquema de conexión.



A título de ejemplo reproducimos, abajo a la izquierda, el esquema de un integrado que contiene 4 circuitos AND.

A pesar de sus reducidas dimensiones, utilizar tales circuitos para la fabricación de un ordenador daría lugar a máquinas gigantescas. Con el desarrollo de la tecnología ha sido posible producir circuitos miles de veces más pequeños, hasta el punto de integrar en un único bloque todos los circuitos necesarios para realizar las principales funciones de un ordenador: dicho circuito se denomina **microprocesador** (μ **P**) y constituye el «corazón» de los ordenadores.

Circuitos compuestos

Los circuitos citados pueden conectarse entre sí para obtener cualquier tipo de función; la conexión entre los diversos circuitos no plantea problema alguno, siempre que los integrados sean de la misma familia.

A modo de ejemplo, veamos un circuito, realizado con integrados AND, que efectúa la función OR (gráfico superior de la pág. 98).

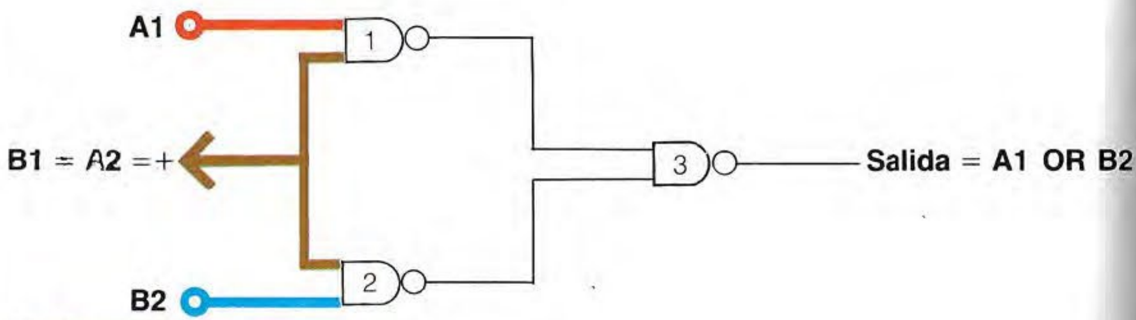
La comprobación de su funcionamiento es inmediata, haciendo la tabla de la verdad y observando que los circuitos 1 y 2 tienen ambos una de las entradas (B_1 y A_2 respectivamente) siempre en el nivel **H**.

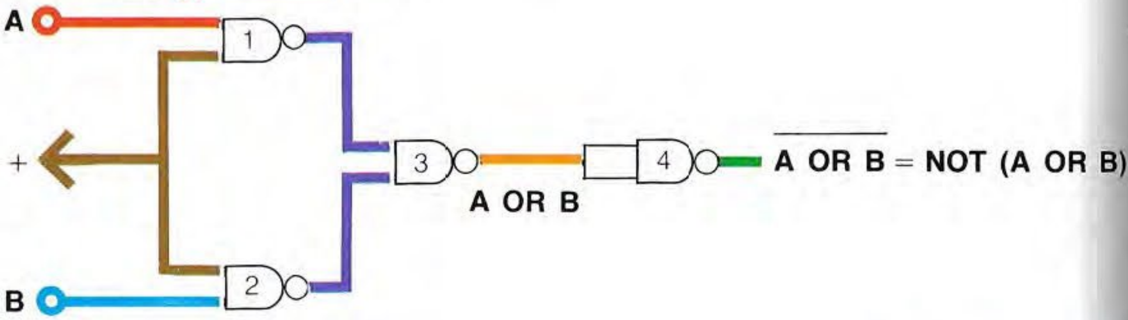
Recordemos que en estos circuitos el nivel **H** puede ser el positivo de la alimentación y el nivel **L** el negativo. Por lo tanto, en el esquema, las entradas B_1 y A_2 están conectadas al positivo de la alimentación. Conectando en la salida otro AND se obtiene la función NOR (pág. 98, gráfico central). Las conexiones a efectuar para obtener las funciones OR y NOR se muestran en el gráfico inferior de la pág. 98.

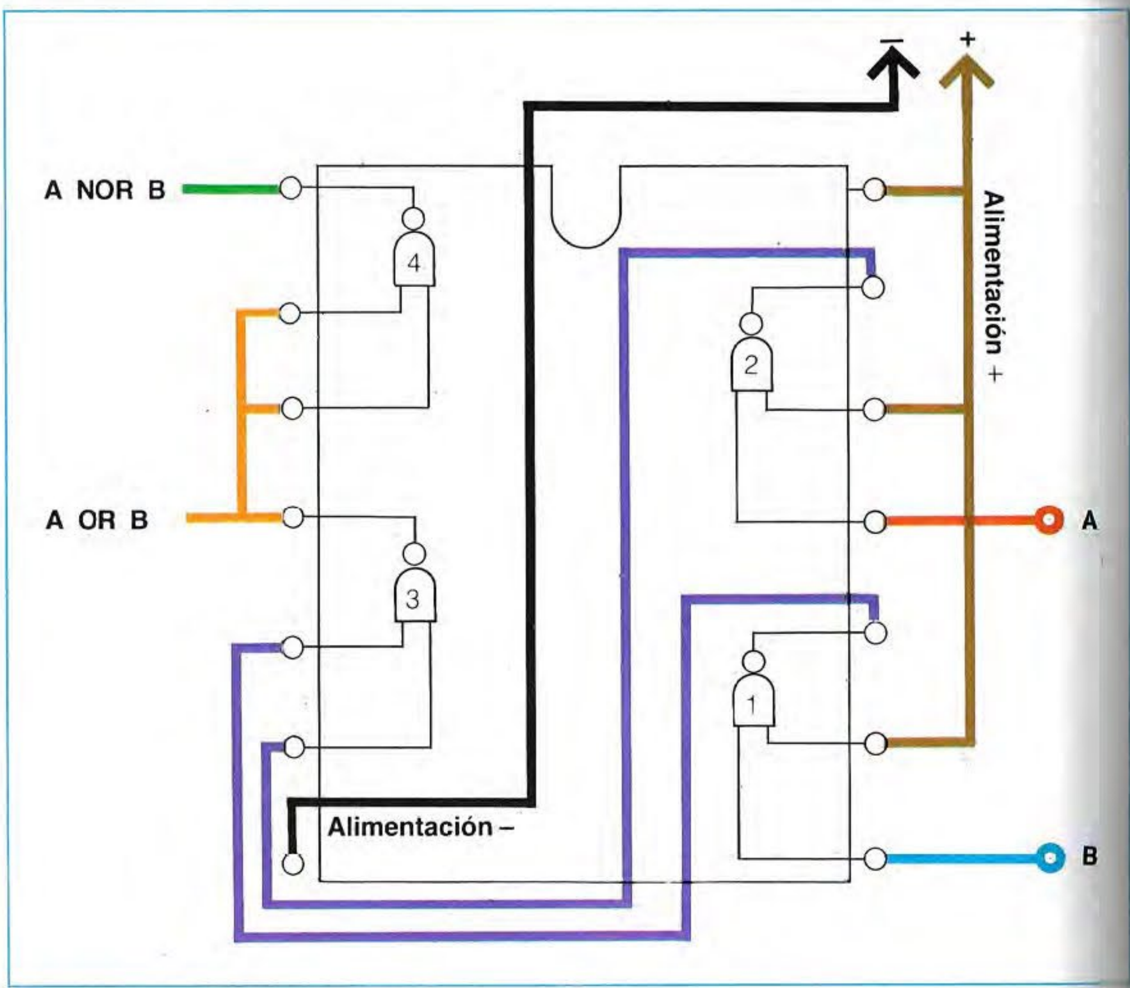
Circuito integrado.



Archivo Curcio/Buffetti Data


A1
1
B1 = A2 = +
3
Salida = A1 OR B2
2
B2


A
1
+
3
4
A OR B = NOT (A OR B)
A OR B
2
B


A NOR B
4
A OR B
3
Alimentación –
2
1
Alimentación +
A
B

Aplicaciones

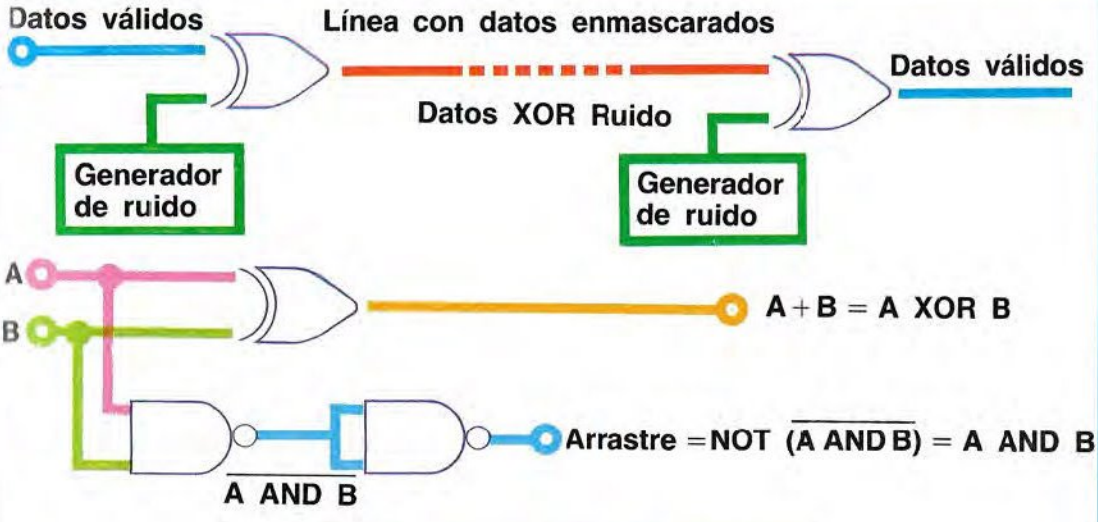
Como hemos visto, la función principal de estos circuitos, en los ordenadores, es la de separadores. Examinemos ahora dos aplicaciones concretas.

La primera se refiere a la seguridad en la transmisión de datos; efectivamente, en las redes de ordenadores existe la posibilidad de efectuar conexiones a larga distancia, y los datos transmitidos pueden ser interceptados, por lo que deben adoptarse medidas de seguridad.

Existe una gran variedad de tales medidas, con distintos grados de sofisticación y seguridad; una de las más sencillas, pero de todos modos suficientemente segura, consiste en enviar por la línea de transmisión, además de los datos, un RUIDO formado por una serie de impulsos casuales que, al mezclarse con los datos, los vuelven irreconocibles. A la llegada es necesario insertar un ruido igual al introducido en origen (operación que sólo puede realizarse si se dispone del mismo generador de ruido) y la señal resultante suministra los datos válidos.

El circuito capaz de mezclar las señales en origen y separarlas en destino es un simple XOR. El esquema puede verse en la parte de arriba del gráfico inferior. Supongamos que en origen el dato sea un nivel **H**; si se introduce como ruido otro nivel **H**, el resultado será un nivel **L** (**H** XOR **H** = **L**), y en destino no se podrá saber si este nivel **L** es un verdadero nivel **L** o un nivel **H** enmascarado, más que introduciendo el mismo ruido de origen (**H**); el resultado será: **H** (ruido) XOR **L** (dato recibido) = **H** (dato verdadero).

La segunda aplicación es un circuito para realizar un sumador binario. En la parte de abajo del gráfico vemos el esquema y la **Tabla de la Verdad**. Obsérvese que al haber utilizado circuitos NAND, que son el tipo más común, han sido necesarios dos, el segundo sólo como inversor. Si en lugar del circuito 1 se hubiese utilizado un AND, no habría hecho falta el segundo, puesto que la salida habría sido directamente A AND B.



A	B	Suma	Arrastre
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Otros tipos de circuitos integrados

Los circuitos integrados que pueden utilizarse en la fabricación de un microordenador no se reducen a los tipos elementales vistos hasta ahora. Hay una gran variedad de ellos, con funciones más o menos complejas y con distintos grados de especialización.
Los circuitos que cumplen la función de operadores lógicos (AND, OR, XOR, etc.) son los más sencillos y de uso más general; al aumentar la especialización del circuito, aumenta también su complejidad, hasta llegar al microprocesador, que en un único integrado reúne miles de componentes.
A continuación, describiremos someramente algunos circuitos más complejos de uso general. Los circuitos específicos para microordenadores serán descritos más adelante.

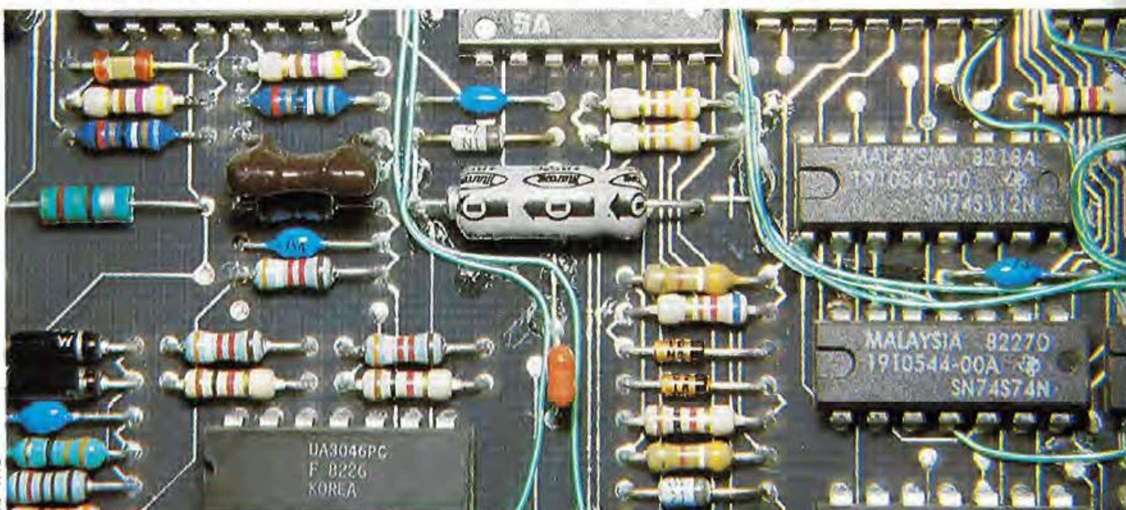
Timer: tienen por objeto generar formas de onda repetitivas en el tiempo; se utilizan generalmente para la creación de los impulsos. Se dividen en dos tipos fundamentales: de red RC y de cuarzo.
En los circuitos de red RC, la duración del impulso es regulada mediante un circuito compuesto por una resistencia y un condensador: son los más baratos y los de menor precisión. Los circuitos de cuarzo utilizan un cristal de cuarzo cuya vibración mecánica regula la duración del impulso generado: son más caros (a causa del cristal), pero su precisión es mucho mayor. Este segundo tipo es el que se utiliza para la generación de señales en los microordenadores. En estos sistemas, la frecuencia es de 2 a 4 MHz (megahertzios).

Trigger: circuitos «gatillo», es decir, sensibles al nivel de la señal de entrada. Cuando dicho nivel supera un determinado valor, «saltan» y la salida cambia de estado (del nivel bajo pasa al alto o viceversa). El tipo más común es el TRIGGER DE SCHMITT. Se utilizan principalmente como regeneradores de señales.

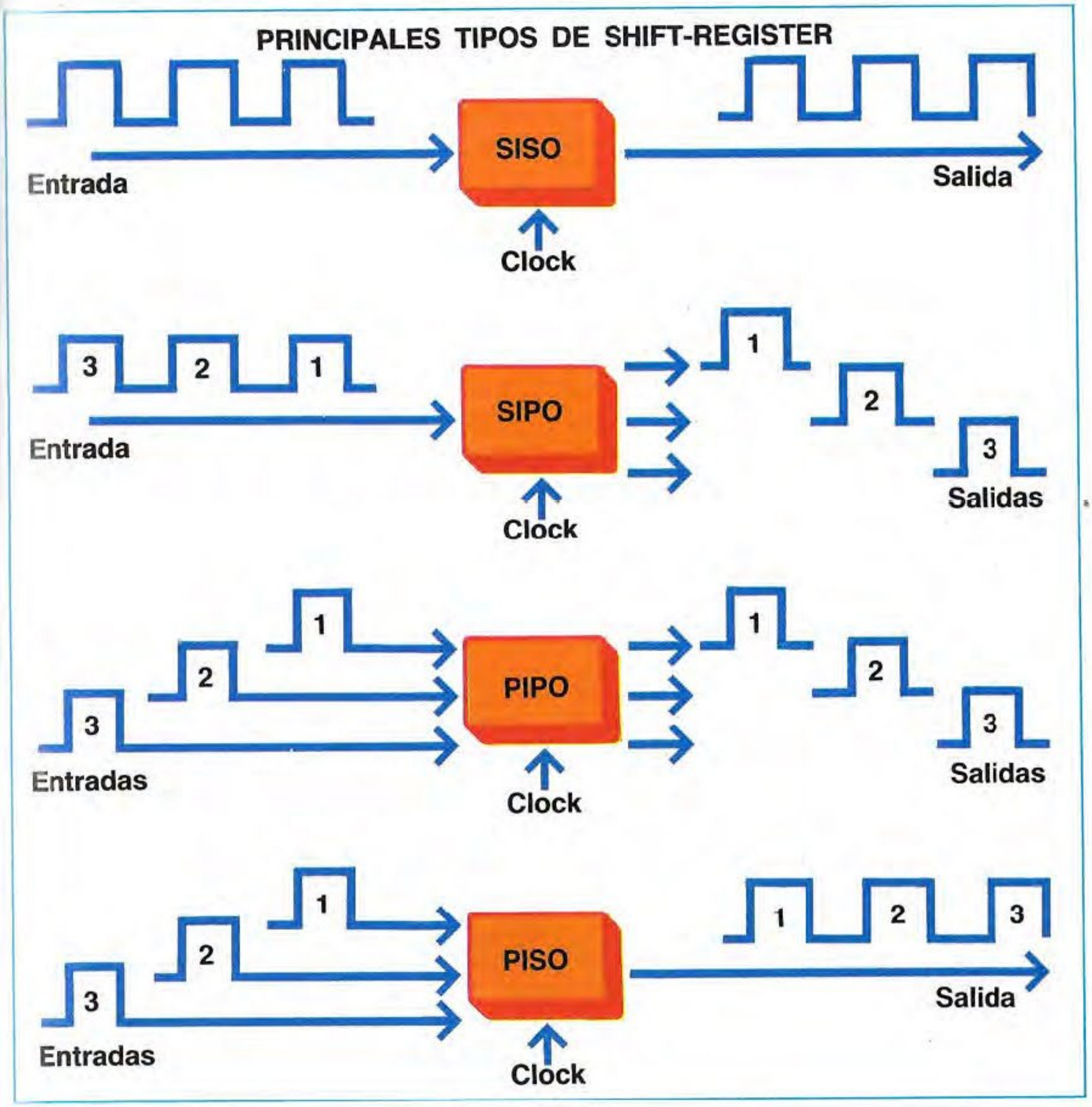
Divisores: pueden suministrar en la salida un número de impulsos igual al de entrada dividido por un número cualquiera. Los hay de dos tipos: de división fija y programables. Los divisores fijos suministran, en la salida, el número de impulsos de entrada dividido por un número fijo, que depende de cómo se conecte el circuito. En los divisores programables, por el contrario, se puede plantear cada vez un divisor distinto.

Flip-Flop: circuito de dos estados (ON-OFF) cuya salida no conmuta siguiendo inmediatamente a la entrada, sino que memoriza el cambio de estado de la entrada y lo sigue al llegar otra señal (impulso) denominada CLOCK.
En otras palabras, es un circuito **biestable**, es decir, que tiene dos estados estables, ON-OFF, que siguen a la entrada tras la orden constituida por el impulso de clock. Se utiliza principalmente como divisor por 2 o como memoria. Este circuito es de tipo sincrónico: la conmutación se efectúa en coincidencia con el clock.

Detalle de un ordenador: en la placa se ven varios circuitos integrados.



Centro THC



Shift-Register: formado por un determinado número de Flip-Flop convenientemente conectados. Puede considerarse como una serie de celdillas contiguas en las cuales el dato (impulso) es desplazado, una celdilla tras otra, a la llegada de cada impulso de clock.

Inicialmente, el dato (impulso) se sitúa en la primera celdilla; al llegar el primer impulso de clock, pasa de la primera celdilla a la segunda, y así sucesivamente. Esta clase de circuito se divide en varios tipos, según su forma de funcionamiento.

Los principales tipos son:

SISO (Serial Input Serial Output):
los datos se introducen en serie, y son presentados, también en serie, en la salida, coincidiendo con los impulsos de clock.

SIPO (Serial Input Parallel Output):
un determinado número de impulsos de entrada es introducido en serie y luego presentado en igual número de salidas distintas. Por ejemplo, 4 impulsos de entrada son memorizados y presentados a la vez en 4 salidas distintas.

PIPO (Parallel Input Parallel Output):
tanto la entrada como la salida pueden aceptar varios impulsos simultáneamente.

PISO (Parallel Input Serial Output):
la entrada acepta varios impulsos simultáneos, que son memorizados y presentados uno tras otro, es decir, serialmente, en una única salida.

Códigos secretos contra los piratas del software

La rápida y universal difusión de las técnicas de transmisión de datos propia de la telemática ha reducido enormemente los tiempos necesarios para que personas u oficinas que se encuentran a miles de kilómetros de distancia puedan intercambiar información, datos y documentos. Los datos confidenciales memorizados en los archivos electrónicos, relativos, por ejemplo, a cuentas bancarias o secretos industriales, son introducidos diariamente en las redes de telecomunicaciones. El número de datos transmitidos se incrementa día a día, y cualquiera podría interceptarlos si no se tomaran las oportunas medidas de seguridad.

Los casos más notorios, que han dado mucho que hablar a la prensa mundial, tienen que ver con la denominada «piratería del software».

Personas tan ingeniosas como carentes de escrúpulos han encontrado la forma de introducirse en las líneas de transmisión de datos de algunos grandes bancos internacionales, por las que diariamente se intercambian comunicados relativos a las operaciones financieras efectuadas o a efectuar en las cuentas de los clientes.

La gestión automática de las transacciones financieras es sin duda una de las más importantes innovaciones que la informática ha introducido en nuestra época. Las precauciones que se toman en los procedimientos de transmisión son muy variadas y complejas, pero no han impedido a algunos especialistas deshonestos manipular en provecho propio la transferencia de grandes sumas a una cuenta corriente previamente abierta por ellos en una filial.

Las noticias de este tipo han sido silenciadas por los organismos bancarios, a los que tal publicidad no hubiera favorecido; pero en más de una ocasión han llegado igualmente a las primeras páginas de los periódicos, despertando una cierta alarma en la opinión pública.

Pero no es sólo la piratería del software lo que empuja a los especialistas a buscar nuevas formas de «camuflaje» de la información. Las comunicaciones de carácter reservado, que diariamente viajan por las líneas telefónicas de todo el mundo, incluyen también noticias sobre inventos y procesos industriales patentados, o mensajes entre organismos internacionales.

Hoy más que nunca se siente la necesidad de poner a punto nuevos sistemas de codificación, utilizando para tal fin las enormes posibilidades ofrecidas por los ordenadores.

El cifrado es un proceso mediante el cual el texto original se traduce letra a letra, de forma que el mensaje cifrado contenga el texto original pero con la posición de las letras cambiadas (cifrado por trasposición) *o con una serie de letras sustituidas* (cifrado por sustitución). *En la práctica se puede emplear una combinación de ambas técnicas, cada una de las cuales puede aplicarse más de una vez al mismo texto.*

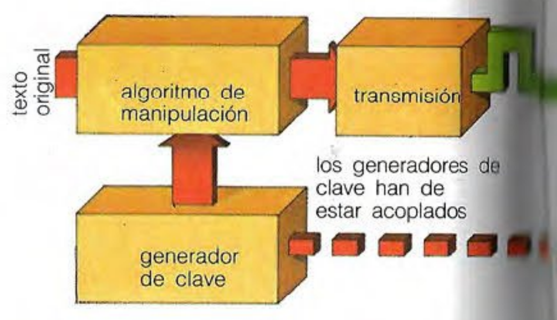
Un cifrado por trasposición ingenioso, en el que todas las letras se mezclen más de una vez (cifrado por trasposición doble o triple), *aplicado a textos relativamente largos, puede constituir un sistema de criptografía válido.*

Un sencillo ejemplo de cifrado por trasposición es el siguiente:

V E N E N
S E G U I
D A C O N
E L M A T
E R I A L

El texto, leído horizontalmente, dice: VEN ENSEGUIDA CON EL MATERIAL, pero el mensaje podría criptografiarse leyéndolo verticalmente: VSDEE EEALR NGCMI EUOAA NINTL. Con un sistema tan simple, un experto se daría cuenta rápidamente de que leyendo, después de la primera, una letra de cada cinco, el texto adquiere sentido. Por lo tanto, hay que adoptar un método de mezcla más complicado. Análogamente, un tipo de cifrado por sustitución muy simple se obtiene corriendo un lugar cada letra del alfabeto, con lo que la A se vuelve B, la B se vuelve C, etc. El mensaje «VEN ENSEGUIDA», cifrado, se

SISTEMA DE CRIPTOGRAFIA SIMPLIFICADO



convertiría en «WFO FOTFHVJEB»[1]. Un ocultamiento de este tipo podría parecer satisfactorio a primera vista, pero la estructura de cada lengua es tal que la solución del problema sería poco más que un juego.

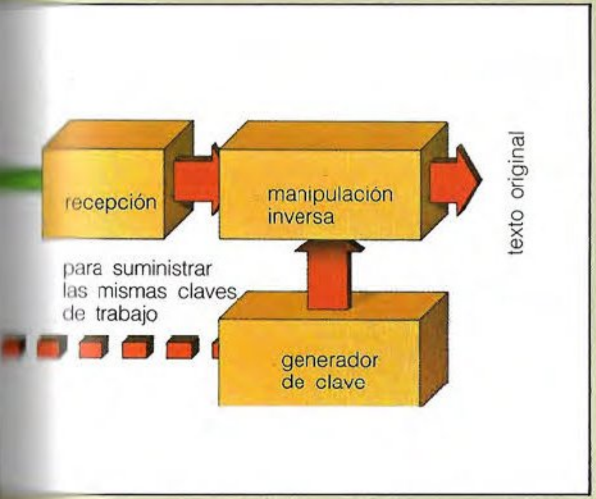
Puesto que en este tipo de enmascaramiento cada letra del texto original está siempre representada por la misma letra del cifrado, éste pertenece a la clase monoalfabética.

En vez de desplazar un solo lugar las letras de la frase VEN ENSEGUIDA, se podría desplazar las sucesivas letras aumentado cada vez en una unidad el número de saltos de lugar, de forma que VEN se convirtiera en WGQ. De esta manera cada letra del texto original puede ser representada, teóricamente, por cualquier letra en el texto cifrado. Con un cifrado de este tipo, descifrar el mensaje exige un análisis matemático más bien complicado.

Con los criptosistemas más complejos hay que seguir dos importantes pasos separados. El primero consiste en desplazar las letras según un esquema preestablecido, que posteriormente podrá repetirse al revés para recuperar el texto original. En la terminología de los ordenadores, en lugar de esquema preestablecido se habla de «algoritmo». El segundo paso consiste en establecer una clave que indique al receptor la forma de aplicar el algoritmo a ese mensaje.

En el caso de los códigos de sustitución, el grado de seguridad del sistema depende en gran medida de la longitud de la clave. Una clave de longitud ilimitada y que incluyera letras y números decimales o binarios sintetizados y dispuestos de forma casual, constituiría, en cuanto a la seguridad, un cifrado realmente inviolable.

1. Para hacer el código más cómodo e internacional, se eliminan del alfabeto la *Ch*, *Ll* y *Ñ* y se añade la *W*.



La mayor garantía de seguridad de un cifrado polialfabético (o de su equivalente en forma digital) estriba en el hecho de que puede utilizar todos los posibles alfabetos sustitutivos. En otras palabras, la letra A, que aparece en el texto en diferentes posiciones, puede ser sustituida cada vez por una letra distinta, para luego ser correctamente descifrada como A por el receptor. Se puede incluso tomar la precaución de que los 26 alfabetos sustitutivos (tantos como letras hay en el alfabeto) se sucedan de manera aleatoria, lo que se logra fácilmente si se dispone de una clave aleatoria de longitud ilimitada.

Este principio es el de la «cadena para usar y tirar»; la cadena es la visualización de una sucesión aleatoria de letras o números decimales o binarios, en la que cada signo clave indica cuántos lugares hay que desplazar cada letra para descifrar el mensaje.

Una criptografía de ese tipo es absolutamente segura y desafía cualquier esfuerzo del criptógrafo por descubrir el código; sin embargo, el sistema plantea problemas al receptor, y resulta arduo y costoso. Efectivamente, hay que producir y distribuir copias de claves de longitud suficiente para cubrir el número total de letras transmitidas durante períodos de semanas o meses.

En la práctica, muchos cifrados polialfabéticos utilizan claves de longitud limitada, basadas en distribuciones tales como los cuadrados de Vigenere o los cuadrados inversos de Beaufort. En ese caso la clave puede ser una frase breve, con claves adicionales suministradas, por ejemplo, en el mensaje mismo. Las técnicas de ataque a los cifrados polialfabéticos se han visto notablemente reforzadas por la disponibilidad de ordenadores de alta velocidad; pero no siempre se necesita o se espera que los sistemas de criptografía sean perfectamente seguros: es suficiente con que retrasen convenientemente la recuperación del texto original, o hagan que la operación de descifrado resulte desproporcionadamente costosa. Un sistema que cumpla este requisito se denomina «a prueba de cálculo».

Durante la pasada década se produjo una notable innovación en la criptografía con la adopción de técnicas numéricas de alta velocidad directamente relacionadas con las que se utilizan en los ordenadores. En electrónica, como ya sa-

bemos, un sistema se denomina digital (o numérico) cuando el número de señales y formas de onda que elabora es restringido, al contrario que los sistemas analógicos convencionales, cuyas señales tienen un número ilimitado de formas y niveles. El sistema binario es un sistema numérico que prevé sólo dos estados distintos, por lo que la señal equivale a una de las dos condiciones «encendido/apagado» (ON/OFF) o, como también se dice en la terminología de las transmisiones, a los niveles «signo/espacio» o «uno/cero». En cierto sentido, todas las transmisiones cifradas por radio o por cable requieren el uso de códigos numéricos. El código Morse y el de los teletipos, por ejemplo, son códigos numéricos, aunque hasta no hace mucho los mensajes eran cifrados como letras antes de ser transmitidos en alfabeto Morse.

La técnica actual consiste en codificar el texto original directamente en forma numérica, usando, por ejemplo, una modulación de impulso codificado. El paso siguiente consiste en cifrar el flujo de cifras binarias (bits) combinándolas con una clave numérica. Gracias a las máquinas electrónicas de alta velocidad, el proceso completo puede efectuarse en tiempo real cualquiera que sea la velocidad deseada, tanto si se trata de un texto escrito como de un discurso hablado, que de este modo puede transmitirse sin demora. En realidad, el empleo de una clave numérica crea una forma de cifrado por sustitución polialfabética. Si la clave es realmente aleatoria y de longitud ilimitada, el sistema ofrece las debidas garantías de seguridad.

Sin embargo, este sistema plantea dificultades operativas similares a las que se presentan en el descifrado analógico con clave «de usar y tirar». En la práctica, se obtiene una clave seudoalcatoria a partir de otra clave relativamente corta (normalmente de menos de 100 bits), pergeñándola de forma que haya largas series de 1 o 0 antes de que se produzca una repetición de la secuencia.

La manera en que se realiza la secuencia seudoaleatoria es, como hemos dicho, muy importante con respecto a la seguridad del criptosistema. Hoy día se sabe que muchos de los sistemas numéricos primitivos, basados en lo que se denomina «lógica lineal», si bien dan lugar a un enorme número de posibilidades de codificación, pueden ser descubiertos rápidamente por ordenadores relativamente pequeños, con tal de que el analista conozca una parte del mensaje (por ejemplo, la dirección de apertura).

Este riesgo ha inducido a varias organizaciones, incluida la IBM, a realizar sistemas criptográficos que, aunque utilizan relativamente pocos bits clave, emplean algoritmos no lineales. Cada bloque de texto y su clave respectiva se someten a una serie extremadamente compleja de cambios, entre ellos la trasposición del orden y las sustituciones basadas en largas secuencias-clave derivadas de unas pocas claves originarias. Estos sistemas se consideran seguros contra todos los ataques, incluso los más enérgicos. En 1977, el American National Bureau of Standards aprobó el algoritmo Data Encryption Standard (DES), basado en una clave de 56 bits. Ca-

Ejemplo 1

CLAVE	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
DESPLAZAMIENTO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

TEXTO ORIGINAL	V	E	N	E	N	S	E	G	U	I	D	A	C	O	N	E	L	M	A	T	E	R	I	A	L
CLAVE DE TRABAJO	C	F	Z	A	L	H	T	R	P	A	S	H	R	Y	F	N	Q	O	B	E	G	R	A	M	N
DESPLAZAMIENTO	3	6	26	1	12	8	3	11	16	1	19	8	18	25	6	14	17	15	2	5	7	18	1	13	14
TEXTO CIFRADO	Y	K	N	F	Z	A	H	R	K	J	W	I	U	N	T	S	C	B	C	Y	L	J	J	N	Z

Ejemplo:
La letra B de la clave cambia C por E, o Y por A
La letra H de la clave cambia C por K, o Y por G

da bloque de datos pasa a través de 18 estadios de manipulación, en los que 16 claves criptográficas diferentes se derivan de una clave principal de 56 bits, de manera que forman una clave global de 10^{17} bits de longitud. Los usuarios del sistema se valen de un algoritmo publicado, pero tienen generadores de claves acoplados mediante los que la clave principal de 56 bits (de hecho, 64 bits con la codificación interna) se mantiene secreta. El generador de claves puede tener la forma de un pequeño módulo sellado, pero lo importante es tenerlo a buen recaudo, lo que conlleva problemas de gestión y de distribución de las claves.

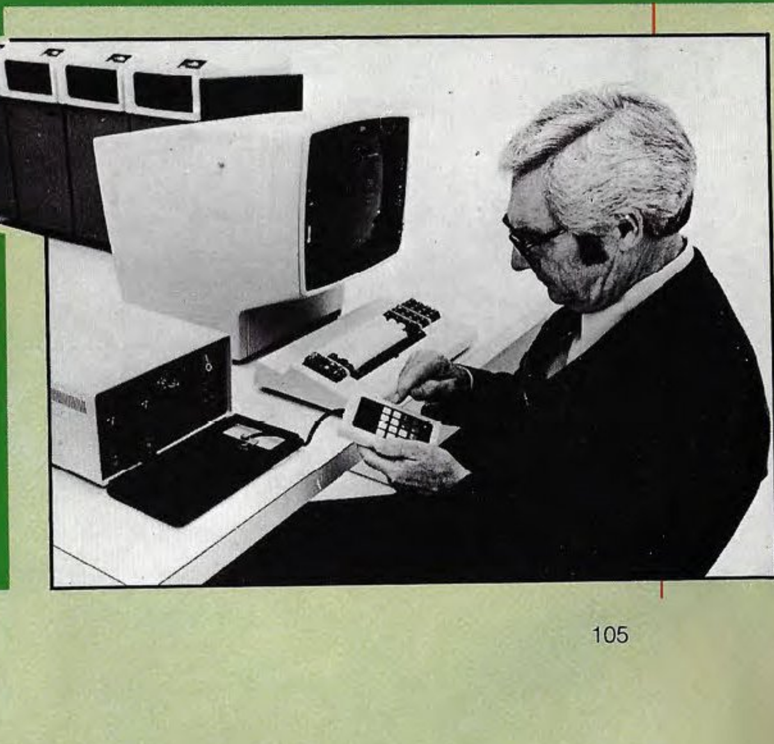
En la actualidad se considera que este sistema es suficientemente seguro para todos los objetivos comerciales normales. Los sistemas numéricos como el DES pueden aplicarse directamente a los ordenadores para evitar la llamada «piratería del software», así como para impedir el acceso no autorizado a archivos confidenciales, o a las transmisiones telefónicas y por radio de datos y mensajes; pero algunos expertos piensan que la clave principal de 56 bits es demasiado corta para que el sistema pueda considerarse absolutamente seguro.

Si bien hoy día los sistemas criptográficos DES se hallan en vías de adopción por parte de organizaciones bancarias y comerciales, y aunque nadie hasta ahora haya demostrado públicamente que pueden ser violados, varios grupos de expertos norteamericanos han propugnado recientemente un sistema alternativo que parece ser que ofrece mayor seguridad: la criptografía con clave pública. Este sistema eliminaría, además, los problemas de gestión y distribución de las claves. En sistemas de este tipo, las funciones de codificación y descifrado están separadas. El remitente no posee la clave de descifrado, sino que cifra el mensaje para un destinatario específico utilizando una lista conocida de claves para la codificación.

Ejemplo 2

A	00001	N	01110
B	00010	O	01111
C	00011	P	10000
D	00100	Q	10001
E	00101	R	10010
F	00110	S	10011
G	00111	T	10100
H	01000	U	10101
I	01001	V	10110
J	01010	W	10111
K	01011	X	11000
L	01100	Y	11001
M	01101	Z	11010

TEXTO ORIGINAL	V	E	N	E	N	S	E	G	U
TEXTO TRADUCIDO AL SISTEMA BINARIO	10110	00101	01110	00101	01110	10011	00101	00111	10101
CLAVE NUMÉRICA	01100	10100	01101	11110	11010	01011	10010	01111	11001
TRANSMISIÓN CODIFICADA	11010	10001	00011	11011	10100	11000	10111	01000	01100
FORMA DE ONDA									



IBM

TEST 2

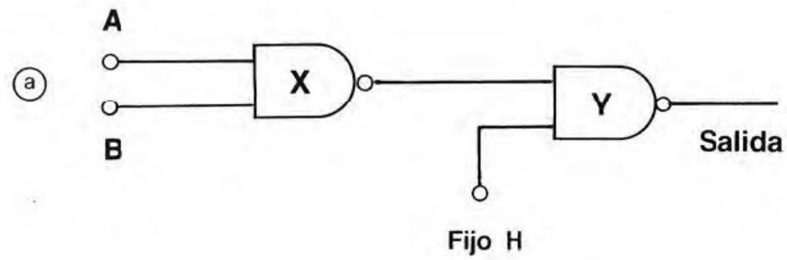
1 / Completar la siguiente **TABLA DE LA VERDAD:**

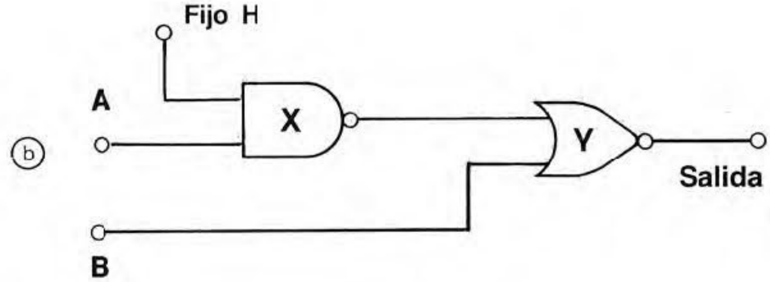
A	B	A XOR B	A AND (A XOR B)
1	0		
1	1		
0	0		
0	1		

2 / Resolver las siguientes operaciones:
a) 11011 AND 011
b) 12 AND 7 (siendo 12 y 7 números decimales)
c) 12 AND 7 (siendo 12 y 7 números octales)
d) 1101 OR 0110
e) 1011 XOR 1101

3 / Dibujar los símbolos gráficos y las tablas de la verdad de los circuitos:
a) AND b) OR c) XOR d) NOT

4 / Escribir las tablas de la verdad para los circuitos:





Las cuestiones 3 y 4 no son determinantes a efectos de proseguir la lectura. El tema se puede considerar asimilado si se contesta la cuestión 1 y al menos tres partes de la 2.
Las soluciones, en la página 122.

Los códigos de transmisión de datos

Para que un ordenador pueda funcionar correctamente, es esencial que esté organizado de forma que pueda interpretar los datos e informaciones que se le suministran.
Mientras que la trasposición de un dato numérico en forma binaria es inmediata, los datos no numéricos han de ser pasados antes a una forma comprensible para la máquina.

Esta condición se satisface adoptando un código para la transmisión de los datos.

Bit y Byte

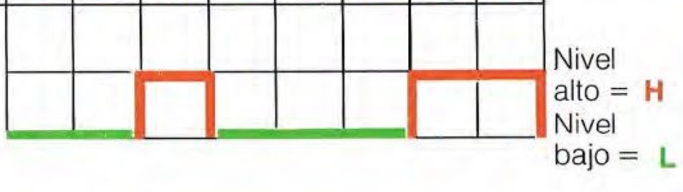
La representación binaria de un número consiste en una serie de símbolos **0** y **1**; cada uno de estos símbolos se denomina **bit**.
Por ejemplo, la notación binaria del número 35 es 0 0 1 0 0 0 1 1 (ver gráfico inferior).
Cada bit es individuado por su posición: así, el bit en la posición 1 vale **1**, el de la posición 3 vale **0**, etc. En el ordenador, cada bit es una señal eléctrica que puede ser de nivel alto (**H**) si el bit correspondiente vale **1**, y de nivel bajo (**L**) si el bit correspondiente vale **0**.

Posición	8	7	6	5	4	3	2	1
Potencia de 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valor	128	64	32	16	8	4	2	1
35 decimal =	0	0	1	0	0	0	1	1

El gráfico muestra una posible situación de las señales correspondientes al decimal 35:

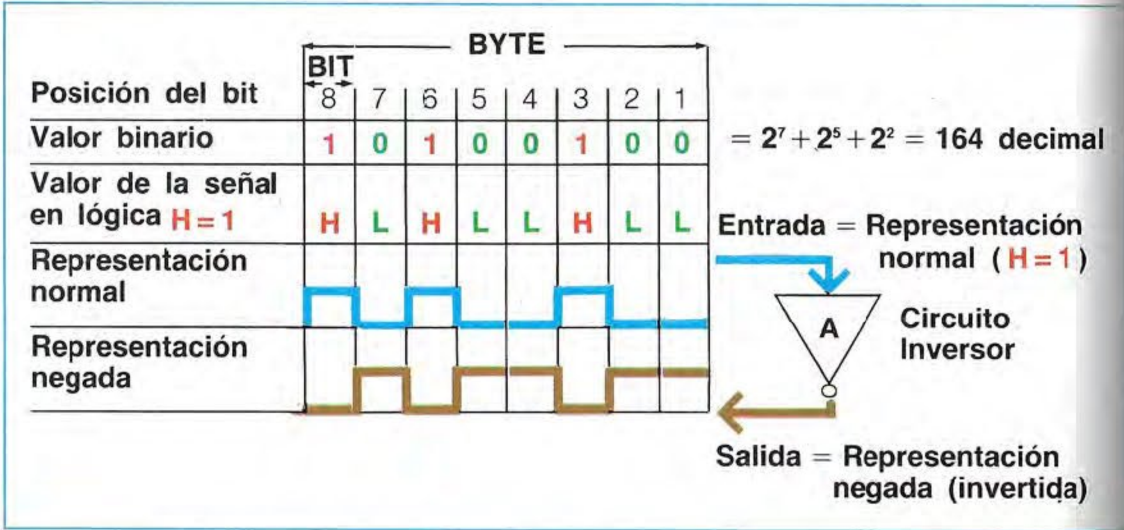
Posición = número del bit	8	7	6	5	4	3	2	1
Potencia de 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valor binario correspondiente a 35 decimal	0	0	1	0	0	0	1	1
Nivel de la señal	L	L	H	L	L	L	H	H

Representación gráfica



En este caso se ha supuesto que el valor binario 1 corresponde a una señal eléctrica de nivel alto; nada nos impide seguir la lógica inversa, es decir, hacer corresponder el valor binario 1 a una señal de nivel bajo. Muchas máquinas funcionan de esta manera (LÓGICA NEGADA).

He aquí la representación del número 164 decimal tanto en lógica normal como negada:



A efectos prácticos, estas dos formas de funcionamiento no plantean ninguna dificultad, y para pasar de la una a la otra basta con un circuito inversor (A en la figura).

Como puede verse en el gráfico, para representar el número 164 decimal en binario se necesitan 8 bits (ya que su equivalente binario es 1 0 1 0 0 1 0 0); un grupo de 8 bits se denomina **byte.**

El máximo valor numérico que puede expresarse con un byte se obtiene igualando a 1 los ocho bits que lo componen:

Valor máximo de 1 byte = 1 1 1 1 1 1 1 1 = $= 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 255$

Con un Bus de Datos de 8 bits (de 0 a 7) el máximo número que la máquina puede utilizar es 255; es, por tanto, inadecuado para las aplicaciones prácticas. Para superar esta limitación hay circuitos capaces de agrupar varios bytes para formar incluso números muy grandes. Por ejemplo, agrupando dos bytes se tienen 16 bits y, por tanto, un valor numérico de cerca de 32.000. Las agrupaciones más empleadas son las de 2 bytes (16 bits) y 4 bytes (32 bits). Obsérvese que la adición de un solo bit equivale a multiplicar por 2 el valor numérico máximo: el valor numérico de un bit cualquiera es igual a la suma de los valores de los valores de los bits precedentes más 1. Por ejemplo, el bit 4 del gráfico inferior vale 16, y es igual a la suma de los valores precedentes más 1 (suma de los bits 0, 1, 2 y 3 = 1 + 2 + 4 + 8 = 15). Análogamente, el bit 7 vale 128, y es igual a la suma de los valores 64 + 32 + 16 + 8 + 4 + 2 + 1 más 1.

Posición = **número del bit**	7	6	5	4	3	2	1	0
Potencia de 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valor	128	64	32	16	8	4	2	1

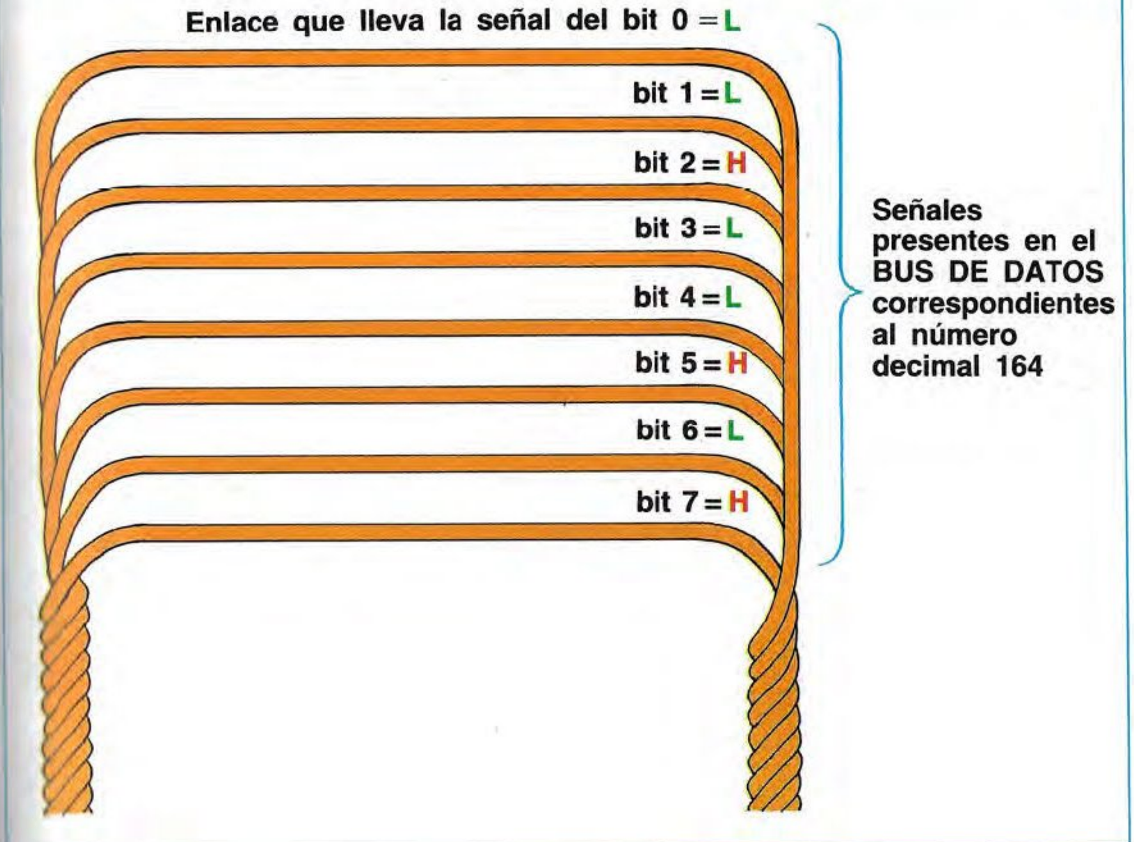
Suma = 15 **+1 =**

Suma de estos valores = 127 **+1 =**

Para representar un número dentro del ordenador, en éste ha de haber tantas conexiones eléctricas como bits componen el número. Para la representación de 8 bits hacen falta 8 conexiones, cada una dedicada a constatar el esta-

do del bit correspondiente. Este grupo de conexiones se denomina genéricamente Bus. En los ordenadores hay dos Bus para dirigir un intercambio de datos: por uno de ellos van los datos y por el otro la dirección de donde se toman o adonde han de llevarse estos datos.

En los microprocesadores, el Bus de Datos es normalmente de 8 bits (1 byte) y el Bus de Direcciones de 16 bits (2 bytes). En el gráfico superior de la pág. 108 la numeración de los bits va del 1 al 8; en la práctica se adopta preferentemente la numeración de 0 a 7 (ver gráfico 2.° de pág. 108), puesto que así el número del bit indica también la potencia de 2 correspondiente. Esta nueva numeración es la que aparece también en el gráfico de esta página, donde se muestra la situación de las señales de un Bus de Datos en el que se ha escrito el número 164.



■ En la notación binaria, la menor cantidad que se puede indicar es un estado o nivel. El valor correspondiente a dicho nivel puede ser 0 ó 1 y se denomina BIT.

■ Un grupo de 8 BITS se denomina BYTE. El máximo valor (decimal) que se puede expresar con un byte es 255.

■ El conductor eléctrico por el que viajan los datos se denomina BUS DE DATOS: en la mayoría de los microordenadores está compuesto por 8 conexiones, y por tanto, puede transferir 8 bits simultáneamente, es decir, un byte.

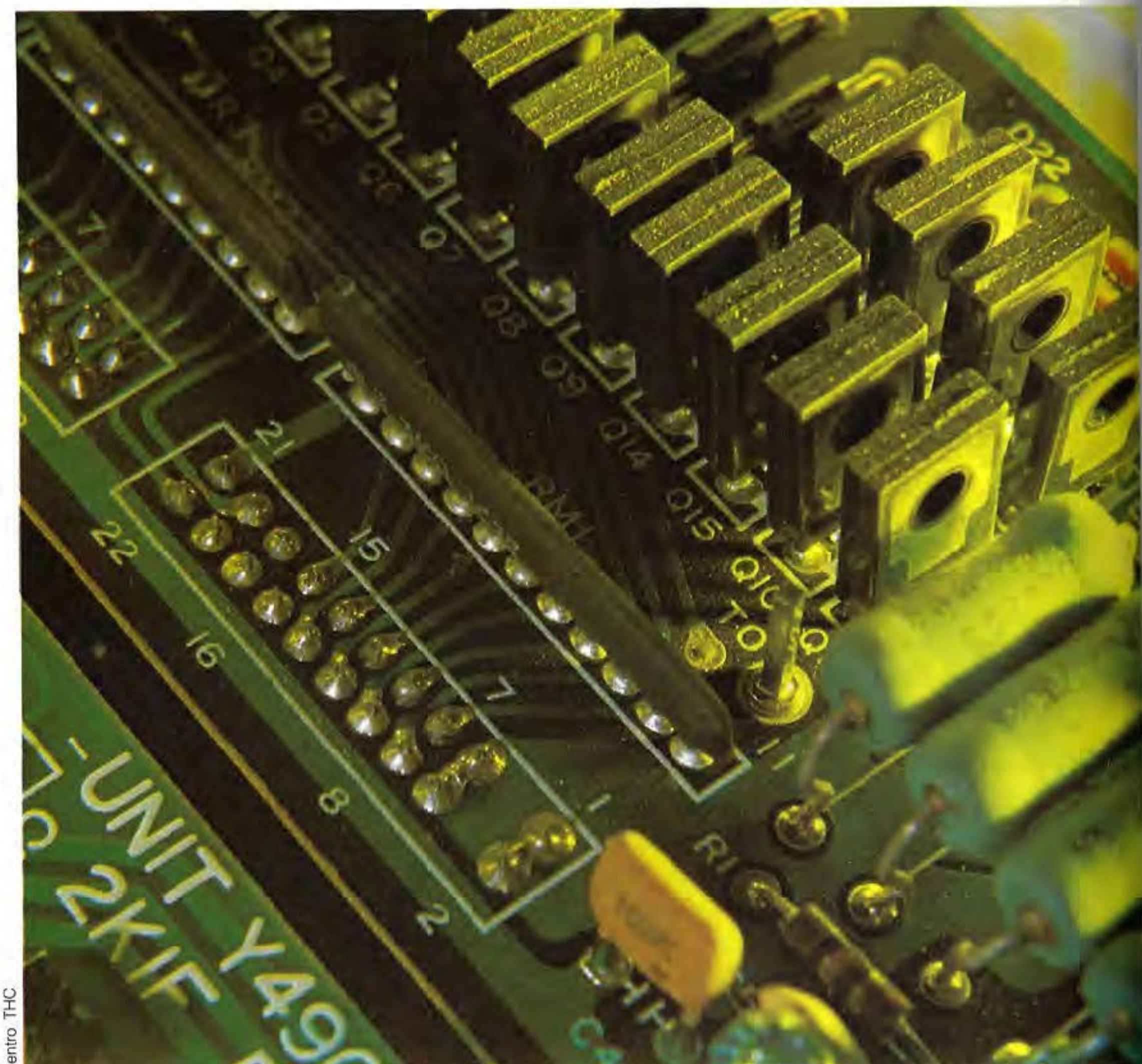
■ Para tomar o depositar un dato, hay que suministrar a los circuitos electrónicos la dirección del dato mismo, es decir, la posición en la que el dato se encuentra.

■ Las direcciones de los datos viajan por una conexión eléctrica denominada BUS DE DIRECCIONES. Este bus suele ser de 16 bits, es decir, 2 bytes.

El Código ASCII

Los ordenadores sólo reconocen los símbolos 0 y 1 (también indicados como L y H). Por tanto, para poder introducir los datos (palabras o números) en la máquina, antes hay que transformarlos en una serie de ceros y unos; viceversa, para leer los datos contenidos en el ordenador se ha de efectuar la transformación inversa, es decir, de una serie de ceros y unos se ha de obtener la equivalente letra del alfabeto o el equivalente número decimal.
La operación de transformación en una serie de ceros y unos se denomina CODIFICACIÓN; la operación inversa es la DECODIFICACIÓN.
La forma más sencilla de codificación es la transformación de un número decimal en su equivalente en notación binaria, es decir, la operación que suministra la representación de un número mediante los símbolos 0 y 1.
Para las letras del alfabeto no existe una forma de codificación «matemática»; la única manera consiste en establecer una tabla de traducción, arbitraria desde el punto de vista conceptual, que a cada letra del alfabeto (u otro símbolo) haga corresponder una serie de ceros y unos. Esta forma de representación se denomina CÓDIGO. Cada letra del alfabeto tiene su propio código, y de esta forma es reconocida por el ordenador.
Viceversa, cuando el ordenador suministra los datos se necesitan circuitos capaces de tradu-

Detalle de los circuitos de control de una impresora.



Centro THC

cir la información binaria en símbolos gráficos comprensibles para el usuario.

La posibilidad de improvisar un código arbitrario en cada caso no es aceptable, ya que, si las tablas de traducción fueran distintas, no podría haber intercambio de datos entre ordenadores sino mediante complicadas operaciones de conversión de los códigos. Por otra parte, un código improvisado no permitiría controlar si el dato ha sido transmitido de la forma correcta.

Por estos motivos se ha establecido un código estándar adoptado por todos los fabricantes: el código **ASCII**.

La sigla ASCII deriva de las iniciales de American Standard Code for Information Interchange. El código ASCII utiliza 7 bits para traducir un símbolo, por lo que se pueden utilizar 128 códigos distintos (de 0 a 1 1 1 1 1 1 1 = 127 decimal).

El código ASCII se divide en tres grupos, cada uno de los cuales posee características y funciones bien determinadas:

- **códigos transparentes**
- **símbolos y números**
- **letras**

Códigos transparentes

Son todos los códigos que no tienen correspondencia en el alfabeto y se utilizan para transmitir instrucciones de control. Durante el intercambio de datos entre el ordenador y la periferia hay que enviar, además de los datos propiamente dichos, una serie de instrucciones para el funcionamiento de las unidades periféricas.

Por ejemplo, al dirigir una impresora el ordenador ha de enviar las órdenes para el cambio de línea y el avance del papel; por lo tanto, la secuencia de las instrucciones de impresión es la siguiente:

1 - Predisposición del cabezal de escritura al comienzo de la primera hoja.
2 - Envío a impresión de la primera línea de caracteres.
3 - Orden, al cabezal de escritura, de regresar al margen izquierdo de la hoja.
4 - Orden de avanzar el papel una línea.
5 - Envío a impresión de la segunda línea de caracteres.
6 - El ciclo vuelve a empezar desde el punto 3 y se repite para todas las líneas a imprimir.

Los códigos enviados a los puntos 1, 3 y 4 son códigos transparentes, puesto que su función es la de controlar la impresora; a su llegada, son reconocidos y enviados a los circuitos electrónicos encargados del control. Los códigos transparentes son 31 y ocupan las primeras posiciones; tienen, pues, valores comprendidos entre 1 y 31 decimal (1 y 1F hexadecimal).

Los principales códigos transparentes se enumeran en la tabla siguiente:

PRINCIPALES CODIGOS TRANSPARENTES

	Sigla	Código decimal	HEX Código hexadecimal
- **Bel** Sonido del timbre del terminal	BEL	7	7
- **Back Space** Desplazamiento hacia atrás de un espacio	BS	8	8
- **Carriage Return** Retorno del carro	CR	13	D
- **Escape**	ESC	27	1B
- **Form-Feed** Salto de página	FF	12	C
- **Horizontal Tabulation** Tabulación horizontal	HT	9	9
- **Line-Feed** Salto de línea	LF	10	A
- **Vertical Tabulation** Tabulación vertical	VT	11	B



Calculadoras: una de ellas, abierta, muestra el circuito interno.

En la tabla se dan también los significados en castellano para facilitar su comprensión, pero normalmente se utilizan los nombres en inglés, de donde derivan las siglas.

Bel (decimal 7, HEX = 7). Muchos terminales disponen de un avisador acústico para llamar la atención del operador. Enviando el código 7 se activa dicho avisador.

Backspace (decimal 8, HEX = 8). Constituye la orden de desplazamiento hacia atrás de un espacio. Normalmente, se utiliza con la pantalla para corregir eventuales errores: se envía la orden BS (código 8) y se superpone el valor correcto al erróneo. Naturalmente, sólo puede utilizarse de esta manera en la pantalla, puesto que el nuevo signo borra al anterior.

Carriage Return (decimal 13, HEX = D). Ordena el regreso a origen del carro de la impresora. En las impresoras esta orden se refiere al carro propiamente dicho, mientras que en las pantallas se refiere al «lápiz electrónico» (CURSOR) que traza los caracteres en la pantalla.

Escape (decimal 27, HEX = 1B). Este código no tiene un efecto visible inmediato. Sirve para informar a los periféricos (impresora o pantalla) de que los caracteres sucesivos son también caracteres de control. Una aplicación típica es la que se da en las pantallas videográficas.
Un terminal que posea pantalla videográfica puede utilizarse también para hacer dibujos: basta con mover el cursor enviando números que indiquen su posición, según el contorno del dibujo a efectuar.
El terminal, si no es informado de que los números que va a recibir corresponden a las posiciones del cursor, los interpreta como caracteres a escribir, con un resultado totalmente distinto. La información acerca de la índole especial de los números por llegar es suministrada por el código ESCAPE.
La forma de la instrucción es:

código ESCAPE + número

Si enviáramos el número sin el código ESCAPE, obtendríamos la escritura del signo correspondiente, en lugar del desplazamiento del cursor.

Form-Feed (decimal 12, HEX = C). Ordena el avance de una página en las impresoras. Constituye una buena medida utilizarlo antes y después de cada fase de impresión, de forma que las distintas listas tengan entre ellas un margen de página en blanco.

Horizontal Tabulation (decimal 9, HEX = 9). En las impresoras y en las pantallas, el máximo número de caracteres que se puede escribir en una línea es constante, por lo que cada signo ocupa una posición bien determinada y la dis-

tancia entre signos es constante. La orden HT permite saltarse un determinado número de dichas posiciones con una sola instrucción. Se utiliza, por ejemplo, para no escribir demasiado cerca del margen: mediante el código 9 se desplaza el carro de la impresora a la primera tabulación horizontal; análogamente, con dos códigos 9 se obtienen dos saltos de tabulación.

Line-Feed (decimal 10, HEX = A). Esta orden permite efectuar un salto de línea.

Vertical Tabulation (decimal 11, HEX = B). Su funcionamiento es análogo al de HT, sólo que el salto de tabulación es vertical. Los códigos HT (tabulación horizontal) y VT (tabulación vertical) pueden ser, en algunos periféricos, programados, es decir, el número de caracteres de una HT y el número de líneas de una VT pueden variarse enviando ulteriores códigos (ver tabla).

TABLA DE LOS CÓDIGOS ASCII TRANSPARENTES

Decimal	Hexadecimal	Sigla	Función en los programas
0	0	NUL	No realiza acción alguna
1	1	SOH	
2	2	STX	
3	3	ETX	
4	4	EOT	
5	5	ENQ	
6	6	ACK	
7	7	BEL	Activa la señal acústica
8	8	BS	Desplaza el cursor un espacio
9	9	HT	Tabulación horizontal
10	A	LF	Salto de una línea
11	B	VT	Tabulación vertical
12	C	FF	Salto de página
13	D	CR	Retorno del carro
14	E	SO	
15	F	SI	
16	10	DLE	
17	11	DC1	
18	12	DC2	
19	13	DC3	
20	14	DC4	
21	15	NAK	
22	16	SYN	
23	17	ETB	
24	18	CAN	
25	19	EM	
26	1A	SU	
27	1B	ESC	ESCAPE
28	1C	FS	
29	1D	GS	
30	1E	RS	
31	1F	US	

Símbolos y números

Este grupo de códigos comienza en el 32 decimal (20 HEX), termina en el 64 decimal (40 HEX) e incluye símbolos y números (ver tabla inferior). En la tabla aparecen también los números hexadecimales, según una nueva forma de representación que considera los números como símbolos gráficos. En los ordenadores se pueden tener dos formas de representación:

- **valor numérico**
- **símbolo ASCII**

La representación numérica se utiliza para realizar cálculos, y la ASCII para las operaciones de impresión. Por ejemplo, el decimal 5 se puede representar en las dos formas:

Numérica (binaria) = 1 0 1
ASCII = 35 HEX = 1 1 0 1 0 1

Cuando se utiliza el 5 como número (para los cálculos) su código es 1 0 1, mientras que en las operaciones de impresión (o sea, en ASCII) es 1 1 0 1 0 1.

TABLA ASCII PARA SÍMBOLOS Y NÚMEROS

Decimal	Hexadecimal	Símbolo o Número Representado	
32	20	Espacio	Símbolos 1.ª parte
33	21	!	
34	22	"	
35	23	#	
36	24	$	
37	25	%	
38	26	&	
39	27	'	
40	28	(	
41	29	)	
42	2A	*	
43	2B	+	
44	2C	,	
45	2D	—	
46	2E	.	
47	2F	/	
48	30	0	Números
49	31	1	
50	32	2	
51	33	3	
52	34	4	
53	35	5	
54	36	6	
55	37	7	
56	38	8	
57	39	9	
58	3A	:	Símbolos 2.ª parte
59	3B	;	
60	3C	<	
61	3D	=	
62	3E	>	
63	3F	?	
64	40	@	

Letras

El último grupo de códigos incluye letras mayúsculas, minúsculas y unos pocos símbolos:

TABLA ASCII PARA LAS LETRAS

Decimal	Hexadecimal	Símbolo
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	
93	5D	]
94	5E	^
95	5F	<
96	60	|
97	61	a
98	62	b
99	63	c

Decimal	Hexadecimal	Símbolo
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	DEL

Circuitos integrados de alimentación. Se los reconoce por las dimensiones superiores a lo normal de los componentes.

Aplicaciones

Veamos la secuencia de los códigos a enviar a una impresora para escribir la frase: «ESTO ES UN EJEMPLO».

El procedimiento constará de los pasos:

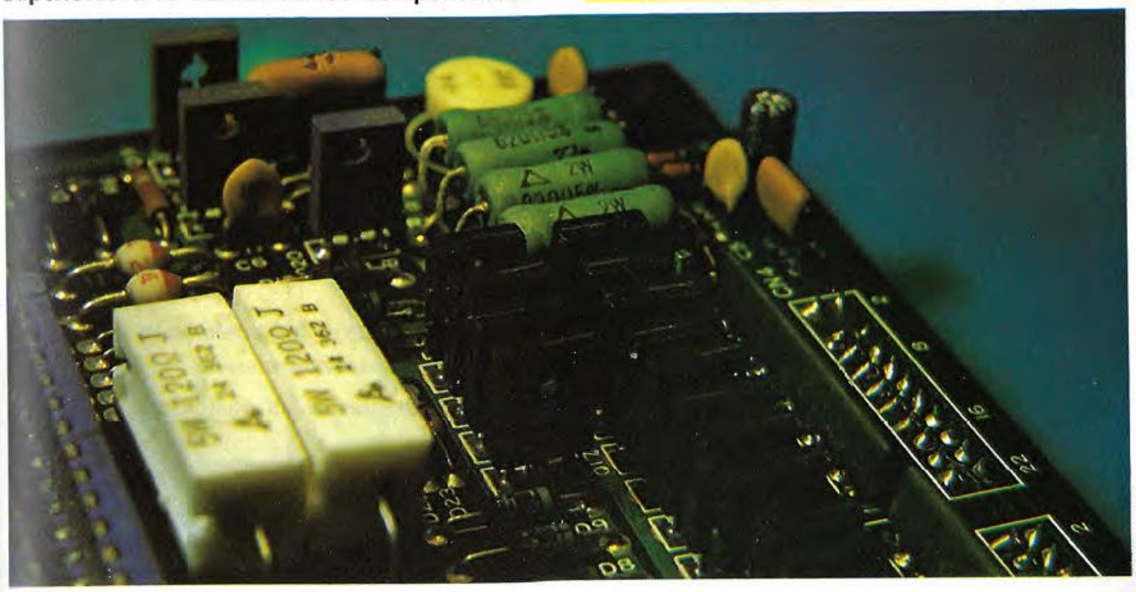
1 - Envío de la orden de posicionado a comienzo de página (FF).

2 - Envío de los códigos correspondientes a la frase que hay que escribir.

3 - Envío de una nueva orden de salto de página (FF).

El código correspondiente a los pasos 1 y 3 es FF = 12 decimal = C (HEX). Para la codificación de la frase hay que proceder letra a letra, de acuerdo con las tablas de conversión adjuntas.

Letra			Código HEX
Q	|	|	51
U	|	|	55
E	E	45	45
S	S	53	53
T	T	54	54
O	O	4F	4F
		espacio =	20
E	E	45	45
\	S	53	2C
		espacio =	20
U	U	55	55
N	N	4E	4E
		espacio =	20
E	E	45	45
S	J	4A	53
E	E	45	45
M	M	4D	4D
P	P	50	50
I	L	4C	49
O	O	4F	4F



Centro THC

FF 12	E S T O E S U N E J E M P L O 45 53 54 4F 20 45 53 20 55 4E 20 45 4A 45 4D 50 4C 4F	FF 12

Tabla ASCII en formato compacto
Damos a continuación la tabla de los códigos ASCII en su forma más compacta y mayormente usada, estructurada de manera que suministre el equivalente binario de cada símbolo:

TABLA CODIGOS ASCII

		000	001	010	011	100	101	110	111
7 6 5 4 3 2 1	COLUMNA / FILA	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	Ø	@	P	\	p
0 0 0 1	1	SOH	DC 1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC 2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC 3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC 4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ET8	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10 (A)	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11 (B)	VT	ESC	+	;	K	[	k	{
1 1 0 0	12 (C)	FF	FS	,	<	L	\	l	¦
1 1 0 1	13 (D)	CR	GS	-	=	M	]	m	}
1 1 1 0	14 (E)	SO	RS	.	>	N	∧	n	~
1 1 1 1	15 (F)	SI	US	/	?	O	—	o	DEL

1 4 2 1 4 2 1

Por ejemplo, el símbolo **F** ocupa la fila 6 y la columna 4.
Para dicho símbolo, el código binario se obtiene uniendo los valores de los bits de la fila con los de la columna:

La fila 6 tiene los valores 0 1 1 0 correspondientes a las posiciones 4, 3, 2, 1

La columna 4 tiene los valores 1 0 0 correspondientes a las posiciones 7, 6, 5

Uniendo ambos números: F = 1 0 0 0 1 1 0 = 70 decimal = 46 hexadecimal

4 6

Este resultado se puede obtener simplemente escribiendo uno tras otro los números de la columna y la fila:

F = columna **4**, fila **6** = **46** hexadecimal

Análogamente, para el símbolo **m**, que se encuentra en la columna 6 y la fila 13 (= D), tenemos:

m = columna **6**, fila **13** (= D) = **6D** hexadecimal

Transmisión de los códigos ASCII

La transmisión de datos entre el ordenador y los periféricos puede efectuarse de dos maneras:

- **transmisión en serie**
- **transmisión en paralelo**

La transmisión en serie consiste en enviar un bit a la vez, uno tras otro, es decir, en serie. Por el contrario, en la transmisión en paralelo se envían varios bits simultáneamente. En la transmisión en serie sólo se necesita una conexión eléctrica, puesto que las señales (bits) van de una en una, mientras que en la paralela hacen falta tantas conexiones eléctricas como señales se envían simultáneamente. En los gráficos de la pág. 118 se esquematizan los dos tipos de transmisión.

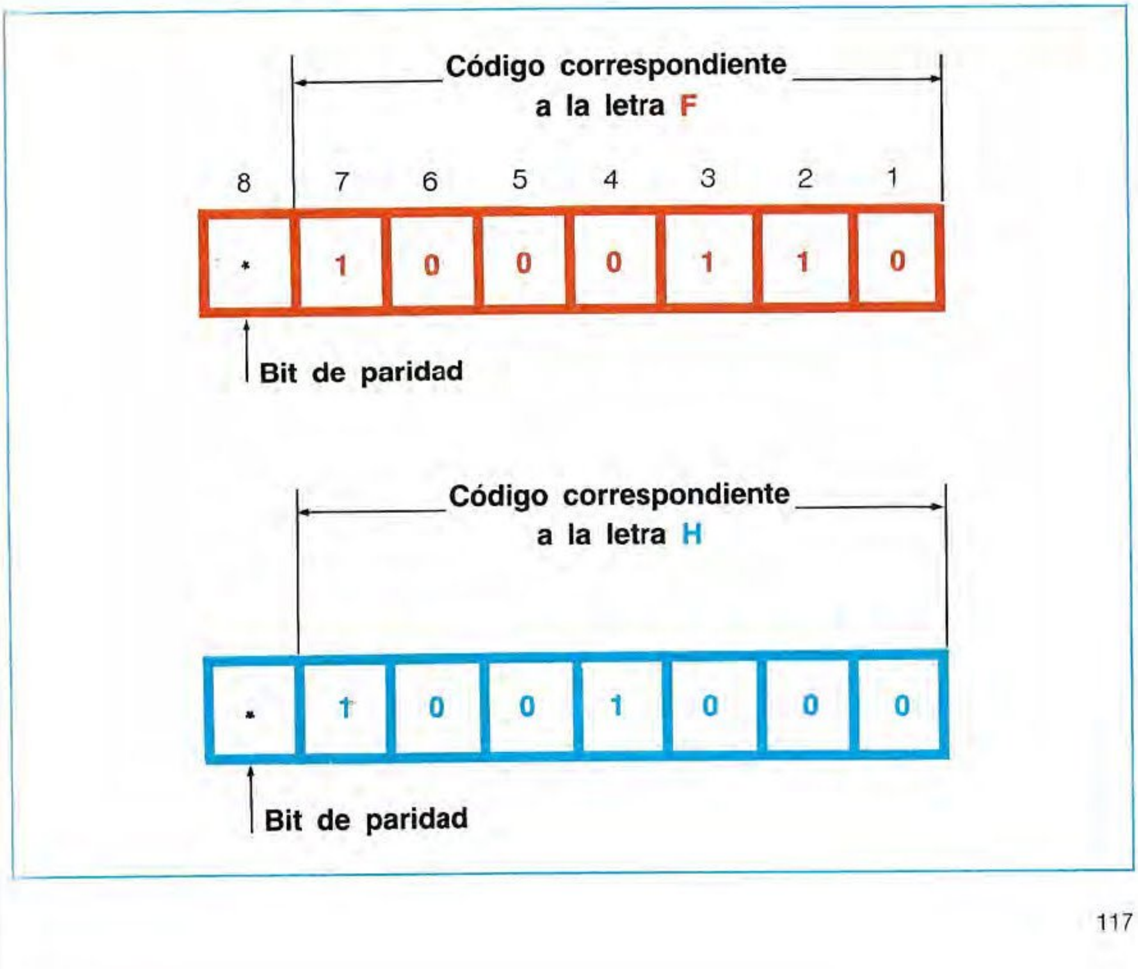
La transmisión en paralelo permite mayores velocidades, pero resulta más costosa y casi imposible de realizar a grandes distancias, debido al elevado número de conexiones eléctricas; el método en serie permite, por el contrario, velocidades más limitadas, pero es más económico y por tanto su uso está más difundido.

De ahora en adelante nos referiremos siempre a la transmisión en serie.

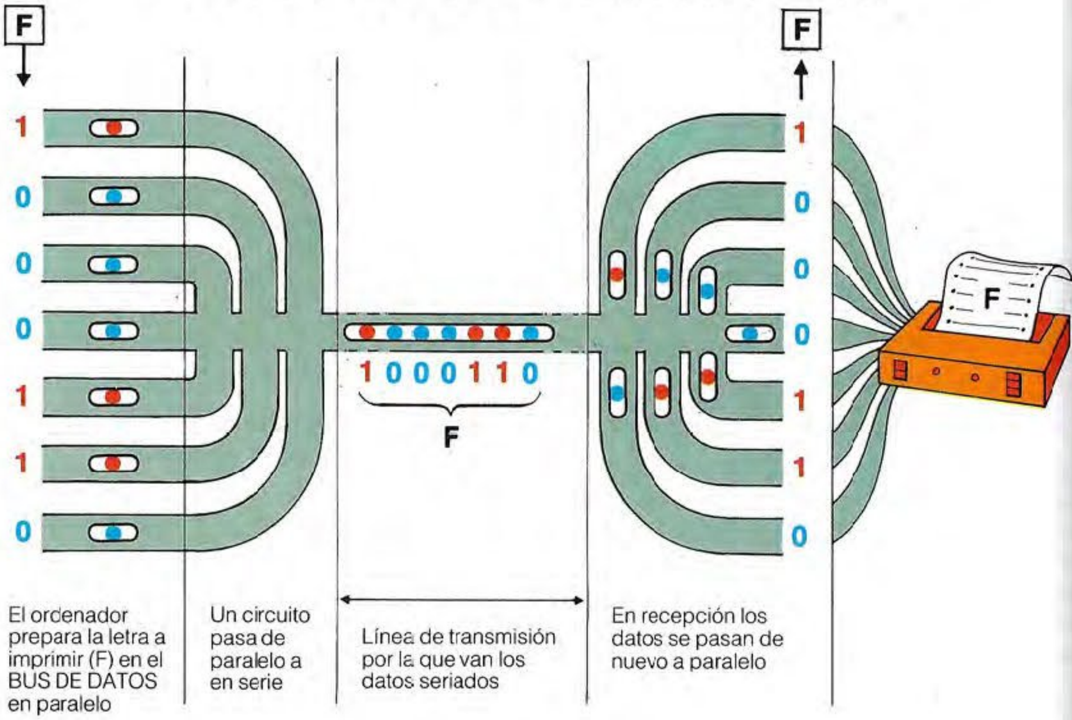
En este tipo de conexión se prevén dos formas distintas de mensaje a enviar:

- **modalidad asíncrona**
- **modalidad síncrona**

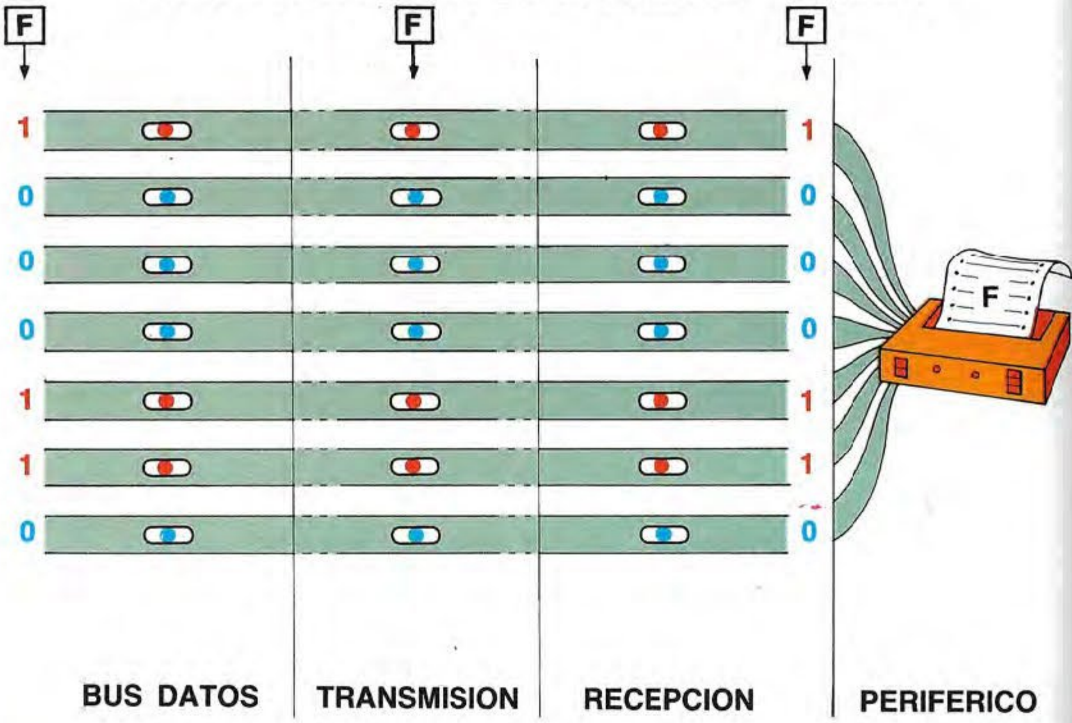
En la modalidad asíncrona la distancia (en términos de tiempo) entre un signo a enviar y otro es arbitraria, mientras que la distancia entre los bits de cada signo es fija. En el gráfico adjunto se esquematiza la transmisión de los signos **F** y **H** en modalidad asíncrona.



ESQUEMA LOGICO DE LA TRANSMISION SERIAL



ESQUEMA LOGICO DE LA TRANSMISION PARALELA



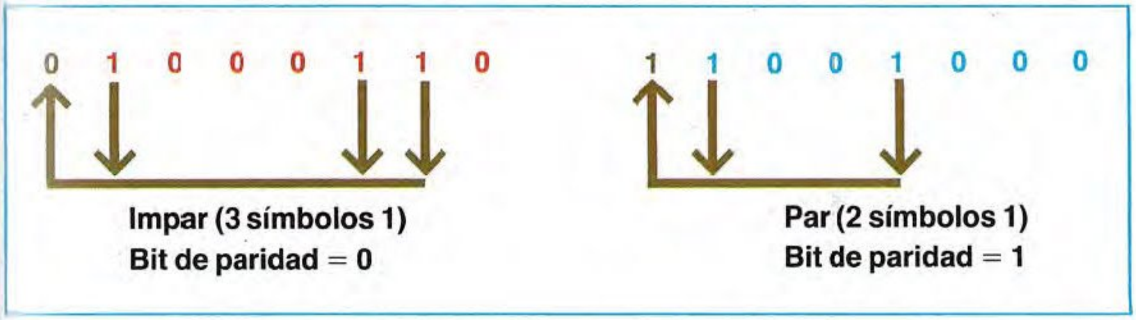


En la modalidad síncrona, los datos han de transmitirse a intervalos de tiempo determinados, que se denominan FRAME; entre un FRAME y otro va, además, un código especial para la sincronización del sistema.

En el gráfico superior se muestra la estructura en la modalidad síncrona para la transmisión de las mismas letras, F y H.

En el gráfico de la pág. 117, además de los 7 bits normales del código ASCII, hay un bit indicado con un asterisco. Este bit, denominado de PARIDAD, es un elemento de control sobre la fiabilidad del mensaje recibido. Este control se efectúa igualando a 1 el bit de paridad (bit 8) si en los otros bits hay un número par de símbolos 1; viceversa, si hay un número impar de símbolos 1 en el signo a enviar, el bit de paridad se iguala a 0.

En el gráfico de la pág. 117 el carácter **F** (1000110) contiene tres 1, por lo que el bit de paridad se iguala a 0; por el contrario, el carácter **H** (1001000) contiene dos 1, y el bit de paridad toma el valor 1. La forma del mensaje será, por lo tanto la siguiente:



En estos ejemplos el bit de paridad vale 0 si el número de símbolos 1 (contenidos en el mensaje) es impar; en la práctica nos podemos encontrar con la lógica inversa, es decir:

bit de paridad = 1 en el caso impar
bit de paridad = 0 en el caso par

Esta inversión es de índole meramente formal.

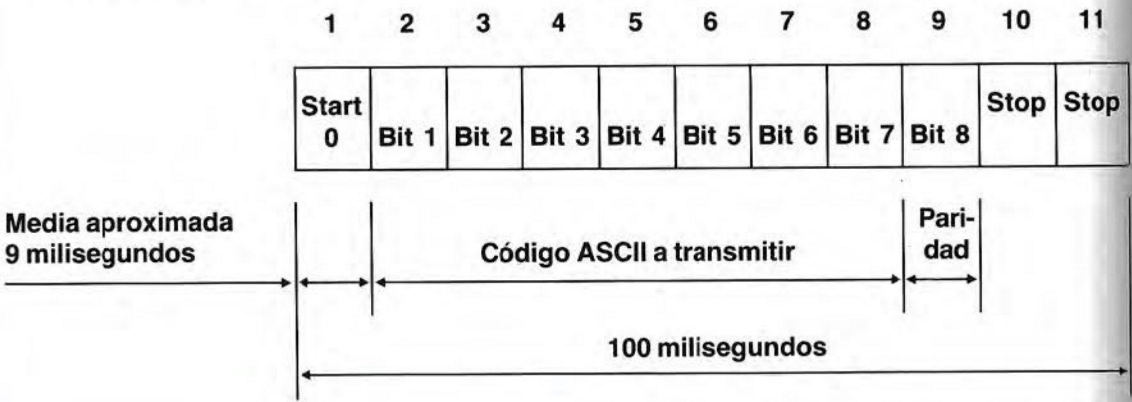
Además del bit de paridad, en la transmisión asíncrona hay que enviar una señal de reconocimiento de comienzo de mensaje y una señal de final de mensaje. La señal de comienzo se denomina bit de START; la de final, bit de STOP. Normalmente los mensajes se concluyen con dos bits de STOP.

El formato completo es, por tanto, el siguiente:

- bit de START
- 7 bits del dato
- bit de paridad
- primer bit de STOP
- segundo bit de STOP

La velocidad de transmisión se expresa en bits por segundo, y su unidad es el BAUDIO. El gráfico adjunto representa un mensaje con una velocidad de 110 baudios. El bit de START vale 0 y los bits de STOP valen 1. El estado 1 se llama MARK, y el estado 0, SPACE.

ESTRUCTURA DE LAS SEÑALES EN TRANSMISION A 110 BAUDIOS



- **Los datos (símbolos, letras, números) se codifican de acuerdo con el código ASCII.**
- **El código ASCII se utiliza tanto en las transmisiones hacia las unidades periféricas (pantallas, impresoras) como en los intercambios entre ordenadores.**
- **Para la gestión de los periféricos hay previstos algunos códigos especiales denominados CODIGOS TRANSPARENTES.**
- **En la transmisión de datos hay dos modalidades posibles: transmisión en SERIE y transmisión en PARALELO.**
- **La transmisión en SERIE es la más usada, y puede ser ASINCRONA o SINCRONA.**
- **En la transmisión ASINCRONA, además de los 7 bits que constituyen el dato se transmiten:
un primer bit de START (que normalmente vale 0);
el bit de PARIDAD, al final de los 7 bits del dato;
dos bits de STOP (que normalmente valen 1).**

El ordenador en las escuelas: joven alumna frente a una impresora en funcionamiento.



Control de recepción

En la transmisión de un carácter de forma asíncrona se añade el bit de paridad como control de la validez del dato. Al recibirlo, hay que controlar la congruencia entre el número de símbolos 1 presentes en el dato y el valor del bit de paridad.

El circuito receptor, al llegar el mensaje, elimina el primer bit (START) y los dos últimos (STOP) y manda a los periféricos (pantalla, impresora) un grupo de 8 bits.

Los 7 primeros bits constituyen el dato propiamente dicho, y el último es el bit de paridad.

Para controlar la validez del dato recibido, se procede del modo siguiente:

1 / se aisla el bit 8 (bit de paridad);
2 / se cuentan los símbolos 1 presentes en los otros siete bits;
3 / si el total es par y el bit 8 es 1, o si el total es impar y el bit 8 es 0, el dato es válido.

Esta serie de operaciones, escrita en una forma comprensible para la máquina, constituye un programa.

Para aislar el bit 8 por una parte y el dato (los 7 bits restantes) por otro, se puede utilizar el operador AND:

valor del bit 8 =
= (mensaje recibido) AND 1
valor de los 7 bits =
= (mensaje recibido) AND (11111110)

Efectivamente, aplicando el operador AND entre un número binario cualquiera (dato recibido) y el valor 1, el resultado es el primer bit del dato en cuestión, es decir, el bit de paridad.

Por ejemplo, en el caso de la letra **F** = 1000110:

```
|      8 bit      |     mensaje recibido
|---------------->| ←  tras quitar el START y
                  |     los STOP
   1000110 | 0
dato = "F" → | ← paridad = 0
                  hay 3 símbolos 1
```

Para controlar el bit de paridad se aplica el operador AND.

Dato AND 1:

```
                 |←  mensaje recibido    →|
dato = F =         1 0 0 0 1 1 0 | 0 bit de paridad
máscara = 1 =      0 0 0 0 0 0 0 | 1
                   -------------------
                   0 0 0 0 0 0 0 |
                       todos = 0 | 0  sigue el bit
                                      de paridad
```

Sin embargo, en el caso de la letra **H** = 1001000:

```
|    mensaje    |
|<------------->|
|1 0 0 1 0 0 0 |1|
dato = "H"      | ← paridad = 1
                |   hay 2 símbolos 1
```

También en este caso para extraer el bit de paridad se aplica el operador AND con la máscara 0 0 0 0 0 0 0 1.

Dato AND 1:

```
dato = H =        1 0 0 1 0 0 0 | 1 bit de paridad
máscara = 1 =     0 0 0 0 0 0 0 | 1
                  -------------------
                        todos = 0 | 1  sigue el bit
                                       de paridad
```

El valor constante con el que se efectúa la operación AND (1, en los ejemplos) con objeto de aislar una parte cualquiera del dato, se denomina MASCARA, precisamente porque permite extraer la parte que interesa leyéndola como si sobre el dato se hubiera puesto una máscara.

Para extraer del mensaje recibido (8 bits) los 7 bits que constituyen la letra transmitida, hay que aplicar la máscara = 1 1 1 1 1 1 1 0.

Mensaje AND 1 1 1 1 1 1 1 0:

```
dato = F =    1 0 0 0 1 1 0 | 0
máscara =     1 1 1 1 1 1 1 | 0
              --------------------
              1 0 0 0 1 1 0 | 0   con esta máscara
               Siguen los          siempre es 0
                 7 bits
                 ASCII
                del dato
```

Los ejemplos anteriores muestran la técnica para aislar los datos y el bit de paridad con los programas adecuados. Los circuitos de recepción modernos la realizan a nivel de hardware, sin necesidad de programa alguno (utilizando, por ejemplo, un integrado AND normal).

Códigos de seguridad

En las tablas de la pág. 78 se listan los resultados obtenidos al aplicar diversos operadores lógicos entre el número constante 11 (máscara) y el numero J variable entre 65 y 90 (decimales). Estos valores son los códigos ASCII correspondientes a las mayúsculas (A = 65; Z = 90).

La columna XOR proporciona un excelente instrumento de codificación, puesto que asocia a cada letra otra letra distinta y no deducible si no se conoce el valor de la máscara.
En la primera línea de la tabla 3 de la pág. 78

11 XOR 65 = 74

la letra A (65) se convierte en J (74).
Cambiando la máscara (tabla 4 de pág. 78)

32 XOR 65 = 97

la letra **A** se convierte en **a** (97).

Este método (que es análogo al de la introducción del ruido) permite obtener una aceptable seguridad en la transmisión de datos. Sin embargo, para burlar los sistemas de seguridad basados en codificaciones, existen métodos consistentes en aprovechar la enorme velocidad operativa de los ordenadores, lo que permite realizar un gran número de intentos hasta encontrar, por aproximaciones sucesivas, la clave del código adoptado.

SOLUCIONES DEL TEST 2

1 / Se plantea el cálculo del resultado utilizando el operador A XOR B, es decir NOT $\overline{\text{(A XOR B)}}$. Para facilitar la solución conviene calcular primero A XOR B y luego su negación:

A	B	A XOR B	$\overline{\text{A XOR B}}$	A AND $\overline{\text{(A XOR B)}}$
1	**0**	**1**	**0**	**0**
1	**1**	**0**	**1**	**1**
0	**0**	**0**	**1**	**0**
0	**1**	**1**	**0**	**0**

Obsérvese que**A** AND ($\overline{\text{A XOR B}}$) es idéntico a A AND B.

2 / a) 11011 AND 11 = 00011

b) Antes de resolver la operación lógica hay que pasar los números decimales 12 y 7 al sistema binario:

12 decimal = 1100;7 decimal = 0111
12 AND 7 = 1100 AND 0111 = 0100

o sea que, poniéndolo todo en decimal (0100 = 4):

12 AND 7 = 4 (en decimal)

c) Los números 12 y 7 octales, pasados a binario son:

12 octal = 001010; 7 octal = 000111

por tanto:

12 AND 7 = 0001010 AND 000111 = 000010

o sea:

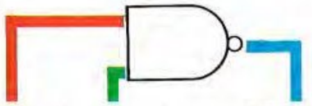
12 AND 7 = 2 (en octal)

d) 1101 OR 0110 = 1111

e) 1011 XOR 1101 = 0110

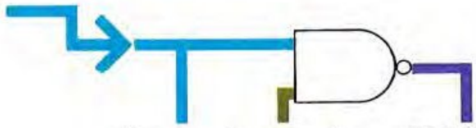
3 / Para la solución, ver los párrafos relativos a los operadores lógicos AND, OR, XOR y NOT, de pág. 80 a pág. 83 y de pág. 88 a pág. 9**1**.

4 / a) En los circuitos compuestos hay que partir siempre del primero, resolver su tabla de la verdad y luego aplicarla al siguiente. De este modo obtenemos, para el circuito X:



A	B	$\overline{\text{A AND B}}$
1	1	0
1	0	1
0	1	1
0	1	1

La columna $\overline{\text{A AND B}}$ (0111) se convierte en una de las dos entradas del circuito siguiente. Este circuito tiene la otra entrada fija en 1; por tanto:



A	B	$\overline{\text{A AND B}}$
0	1	1
1	1	0
1	1	0
1	1	0

b)



A	F	A NAND F	B	B OR (A NAND F)
1	1	0	0	1
0	1	1	1	0
		0	1	0
		1	0	0

En la segunda parte de la tabla se han repetido los estados 0 y 1 de A NAND F para obtener todas las combinaciones posibles con la columna B.

TEST 3

1 / Escribir los números 7, 9, 0 y 3 en binario, como números y como caracteres.

2 / Enumerar las principales diferencias entre la transmisión serial y la paralela.

3 / Escribir el mensaje completo que se ha de enviar para transmitir el carácter K.

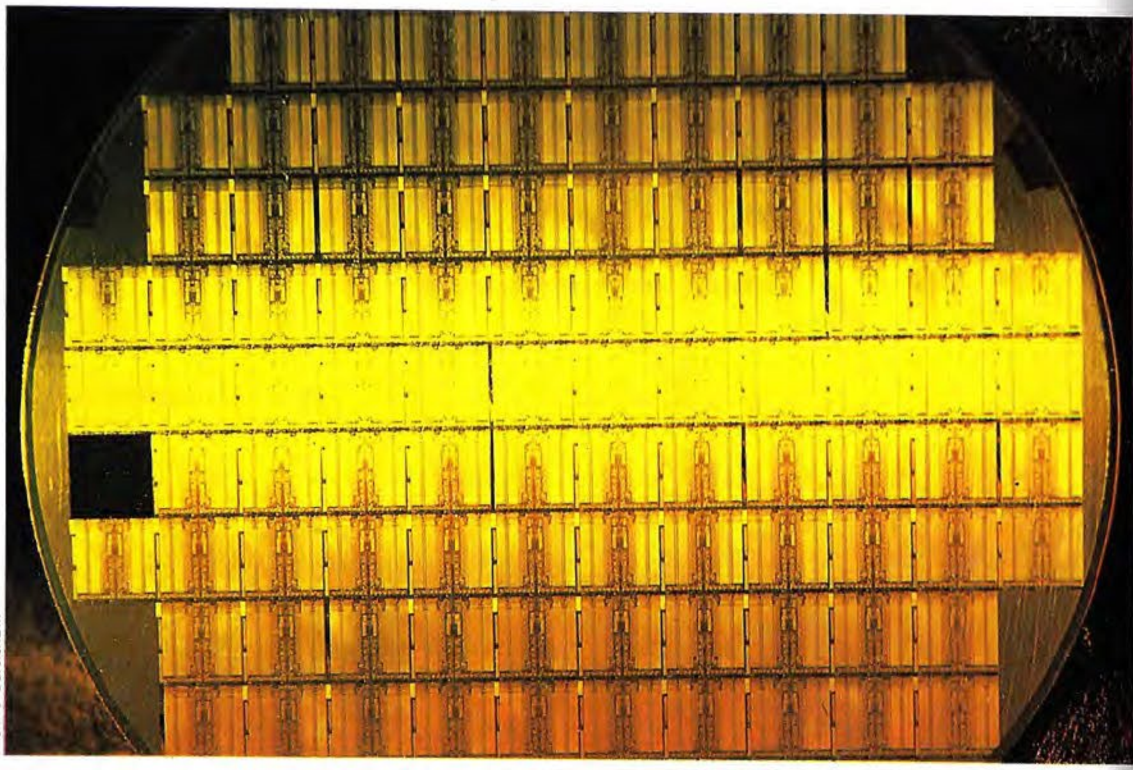
4 / Traducir en códigos binarios las siguientes operaciones a efectuar por la impresora:

a) salto de página
b) escritura de los caracteres X e Y
c) salto de una línea
d) escritura de los mismos caracteres en minúscula

5 / Observando la tabla de los códigos ASCII, encontrar la operación necesaria para convertir cualquier letra mayúscula en minúscula.

6 / Escribir la máscara que extrae de un dato cualquiera los bits 3, 7 y 1 (con los bits numerados a partir de 0). Comprobar la exactitud de la solución con los números decimales 12, 21, 6 y 15.

Las soluciones, en las págs. 150 y 151.



Archivo Curcio/IBM

El dispositivo digital con junta de Josephson permite altas velocidades de conmutación.

Sistemas de microordenador

Para un correcto uso de los microordenadores u ordenadores personales, son necesarios algunos conocimientos básicos sobre la estructura física de dichos aparatos.
Se pueden distinguir tres niveles de utilización de un microordenador:

- **utilización de programas ya preparados**
- **utilización evolucionada, con la posibilidad de modificar los programas según las propias necesidades**
- **desarrollo de programas para nuevas aplicaciones**

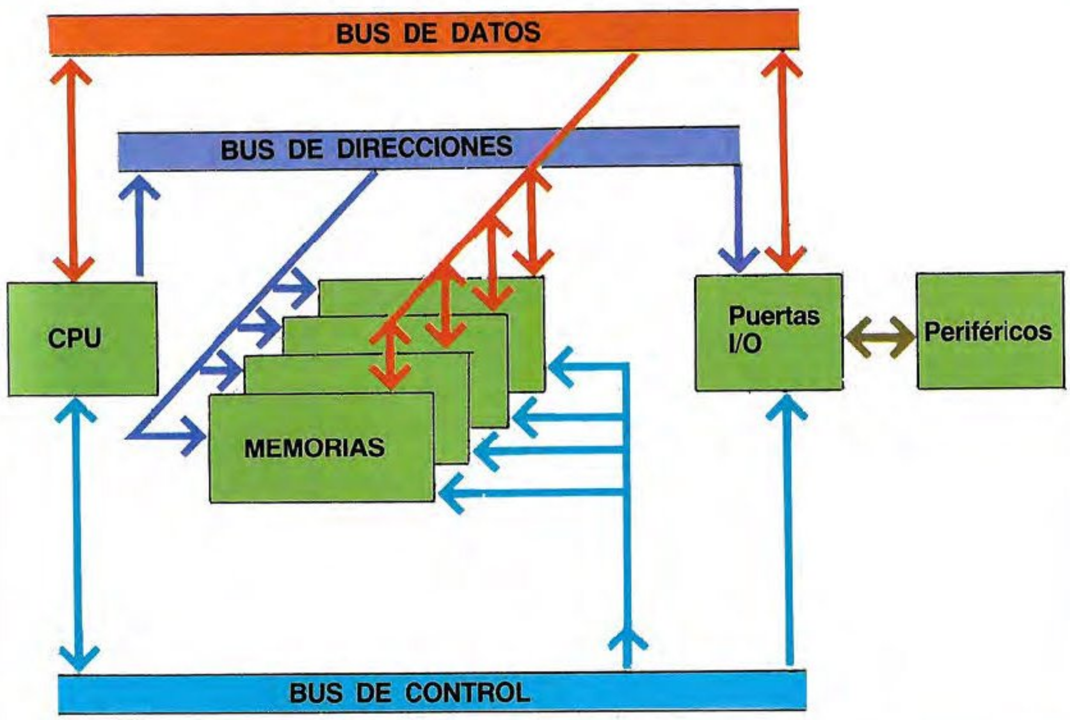
Los conocimientos necesarios están, naturalmente, en función del tipo de utilización, pero, incluso al nivel más sencillo, es conveniente conocer, al menos desde el punto de vista descriptivo, la estructura física de dichos sistemas. Todo lo que respecta a la programación se denomina SOFTWARE, mientras que lo relativo a la estructura física es HARDWARE.
El hardware de los microordenadores es la tecnología que está teniendo mayor desarrollo. Ello permite producir sistemas de costo cada vez más bajo y prestaciones cada vez mayores, hasta el punto de que ya, en algunos casos, los programas valen más que la propia máquina.
Las elevadas prestaciones con bajo costo de hardware hacen posible la utilización de los microordenadores en un número cada vez mayor de nuevas aplicaciones, que irá en aumento en un futuro inmediato.

Estructura básica

En el gráfico de esta página vemos la estructura típica de un microordenador.
Los principales elementos son:

- **CPU = Central Processing Unit (Unidad Central de Proceso)**
- **MEMORIAS**
- **PUERTAS I/O**
- **PERIFÉRICOS**
- **BUS DE DATOS**
- **BUS DE DIRECCIONES**
- **BUS DE CONTROL**

SISTEMA BÁSICO DE MICROPROCESADOR



CPU

Realiza dos funciones principales: rige todos los demás componentes y efectúa los cálculos.

MEMORIAS

Dispositivos que contienen, en forma binaria, las instrucciones que constituyen el programa y los datos que deberán utilizarse durante la ejecución del mismo.

PUERTAS I/O

Circuitos mediante los cuales el sistema puede comunicarse con el mundo exterior. La sigla I/O deriva de las iniciales en inglés de la función realizada: Input/Output (Entrada/Salida).

PERIFÉRICOS

Todos los dispositivos que no forman parte de los circuitos internos.
Los principales tipos de periféricos son:
Video: una pantalla mediante la cual la máquina se comunica con el operador.
Teclado: estructuralmente es muy similar al de una máquina de escribir. Constituye el elemento de entrada de los programas y los datos.
Impresora: el elemento de salida alternativo a la pantalla.
Discos y cintas magnéticos: constituyen la memoria masiva del sistema. Los datos se registran en un soporte magnético y se toman en el momento oportuno. Estos dispositivos no son periféricos propiamente dicho: se pueden considerar como una extensión de la memoria de la máquina.
Dispositivos especiales: en muchas aplicaciones (como el control de proceso en las industrias) se necesitan periféricos especiales capaces de gobernar instalaciones o controlar su funcionamiento. Estos circuitos se fabrican con objetivos muy concretos y no son de uso general.

BUS DE DATOS

Es la conexión física mediante la cual se transmiten los datos. Los sistemas más difundidos utilizan un Bus de Datos de 8 bits (1 byte).
En la práctica también se suele llamar al byte «palabra» u «octeto»; sin embargo, lo recíproco no es correcto: una palabra no está formada necesariamente por un solo byte (en algunos sistemas contiene 16 bits, es decir, 2 bytes).

BUS DE DIRECCIONES

Por estas conexiones van las señales que constituyen la dirección, es decir, la localización de memoria o la puerta I/O en la que han de ser escritos o leídos los datos.

BUS DE CONTROL

Tercera conexión, siempre presente en un sistema, a través de la cual la CPU envía las instrucciones a los demás componentes y recibe de ellos las señales de respuesta. En el gráfico superior de la página contigua se detallan los componentes que constituyen un sistema completo.

Estructura de la CPU

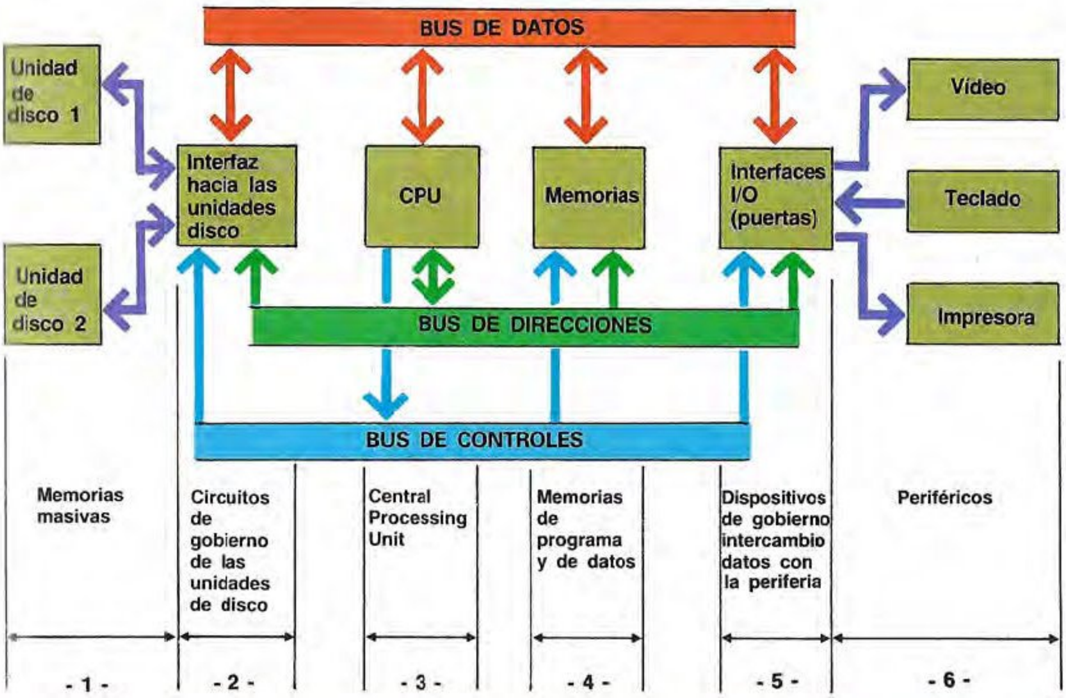
La CPU es el componente más complejo. En su interior se generan todas las señales necesarias para el funcionamiento de los demás circuitos y se realizan todos los tratamientos de los datos. En el gráfico inferior de la página contigua vemos el esquema simplificado de una CPU típica. Como puede verse, su estructura es similar a la del sistema entero.
Sus principales componentes son:
Bus interno: cumple una función análoga a la de los Bus externos (ver gráfico superior), es decir, la de transferir la información.
Acumuladores y registros: en estos componentes se memorizan los datos antes y después del tratamiento.
Decodificador de instrucciones: es el circuito que traduce las instrucciones de un programa en una serie de impulsos eléctricos.
Circuitos de control: son gobernados por el circuito anterior y sirven para generar diversos tipos de señales para el funcionamiento de los demás componentes, sean los internos de la CPU o los externos (memorias, puertas, etc.)
ALU: la sigla de este circuito deriva de las iniciales de su denominación en inglés: Arithmetic Logic Unit (Unidad Lógico-Aritmética). Es la parte destinada a realizar los cálculos y las operaciones lógicas.

Funcionamiento de la CPU

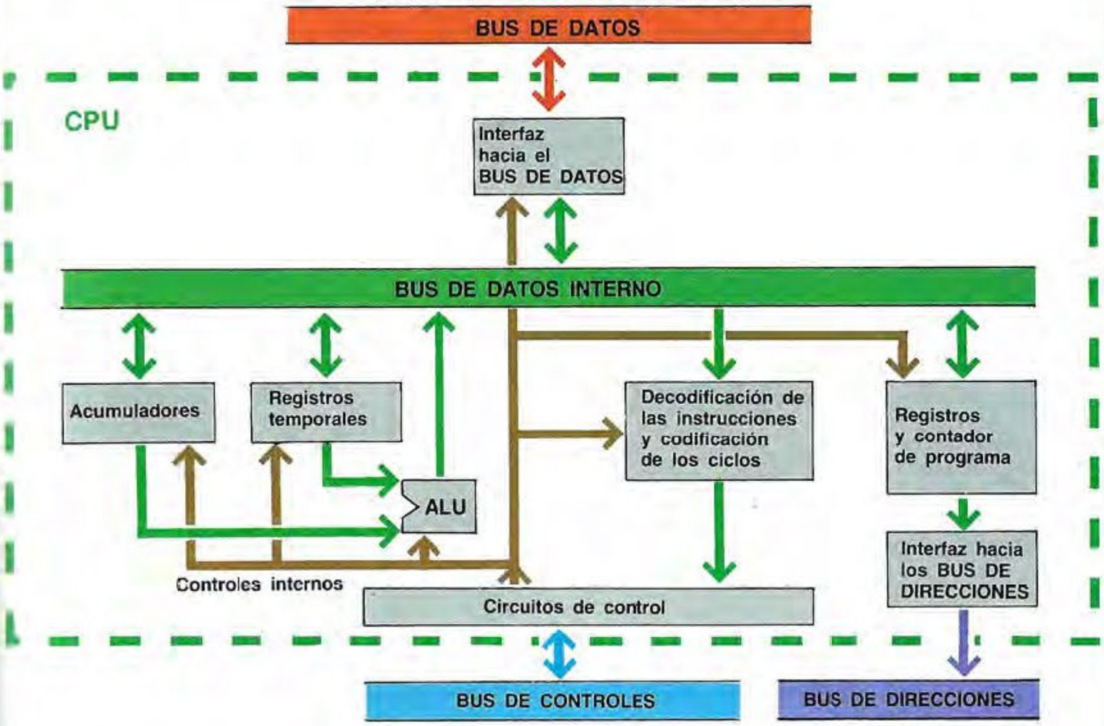
Un programa está compuesto por una serie de instrucciones, cada una de las cuales está dividida en al menos dos partes:

- **Código de la instrucción: expresa la acción a realizar**
- **Dato: la cantidad sobre la que se debe realizar la acción**

SISTEMA COMPLETO DE MICROPROCESADOR



ESTRUCTURA TIPICA DE UNA CPU

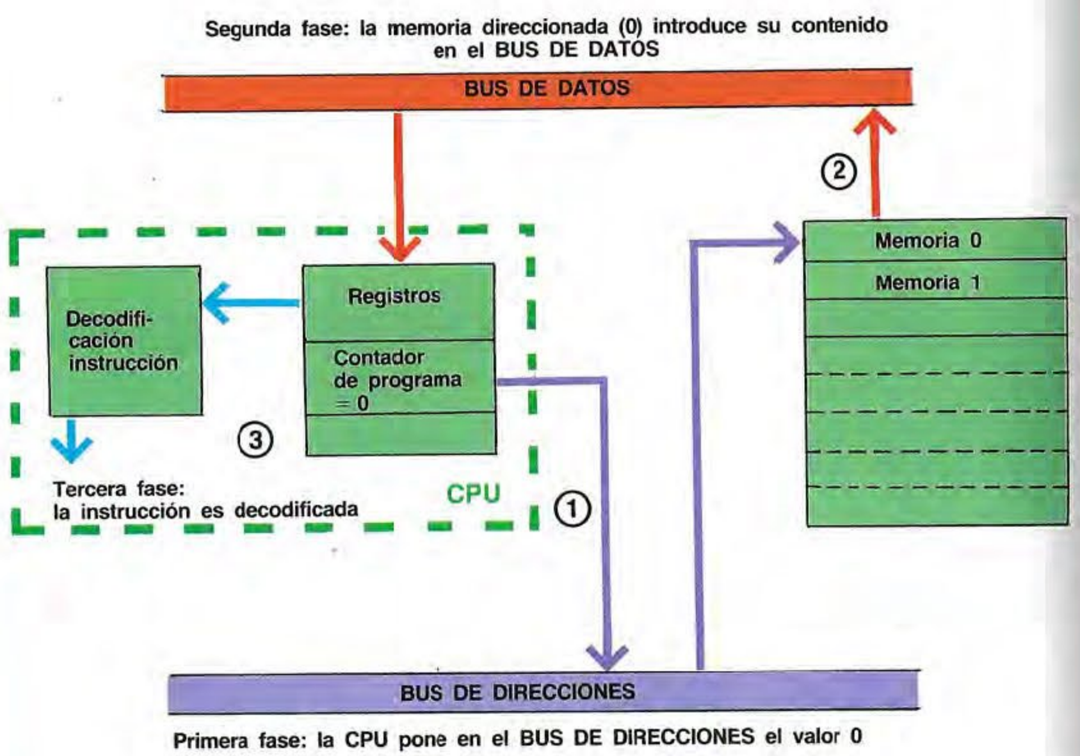


Los códigos y los datos correspondientes (o sus direcciones) se escriben en las memorias del sistema. El conjunto de las memorias puede considerarse como un libro, en cada una de cuyas «páginas» hay escritos un determinado número de caracteres.
Por ejemplo, consideremos una memoria dividida en páginas de 1.024 bytes (ver gráfico de página 125); para conocer el contenido de la memoria 3 de la página 1 hay que dar la dirección 1.026 (la primera página es la n.º 0 y contiene las memorias de 0 a 1.023; la página 1 empieza en la 1.024).
Cada zona de memoria (constituida por una o más páginas) se dedica a una finalidad concreta. La tabla que muestra las diversas zonas y su utilidad se denomina MAPA DE MEMORIA; dicha tabla varía en función del tipo de máquina y de los programas que contiene.
El desarrollo de una instrucción de programa tiene lugar de acuerdo con una secuencia de operaciones denominada CICLO.
En general, las fases en que se articula el desarrollo de un ciclo son las siguientes:

- **búsqueda de la instrucción (FETCH)**
- **decodificación y ejecución (EXECUTE)**

Esta es la forma más simple; instrucciones más complejas requieren varias operaciones.
En el momento de conectar la máquina, o cuando se da la orden de comenzar una tarea, la CPU toma la primera instrucción, la ejecuta, pasa a la segunda y así sucesivamente.
Para seguir esta secuencia, en el seno de la CPU hay una memoria denominada CONTADOR DE PROGRAMA (normalmente indicada con las siglas PC); en esta memoria se almacena la dirección de la instrucción siguiente a ejecutar. Si todas las instrucciones fuesen secuenciales (es decir, si hubiera que realizar un programa sin saltos de un punto a otro), para conocer la dirección de la instrucción siguiente bastaría con sumar 1 al contenido del PC.
En el gráfico adjunto se esquematiza un ejemplo de FETCH (búsqueda de instrucción).

EJEMPLO DE FETCH



① La CPU pone el contenido de PC (en este ejemplo = 0) en el Bus de Direcciones.

② La memoria correspondiente (dirección 0) introduce su contenido (instrucción que la CPU ha de ejecutar) en el Bus de datos.

③ El contenido del Bus de Datos (en este caso una instrucción a ejecutar) se carga en una memoria interna de la CPU (registro); del registro pasa a la unidad de decodificación y es transformada en las correspondientes señales eléctricas. La transformación de la instrucción en las señales se efectúa mediante microprogramas especiales denominados MICROCÓDIGOS, grabados de forma permanente en la CPU.

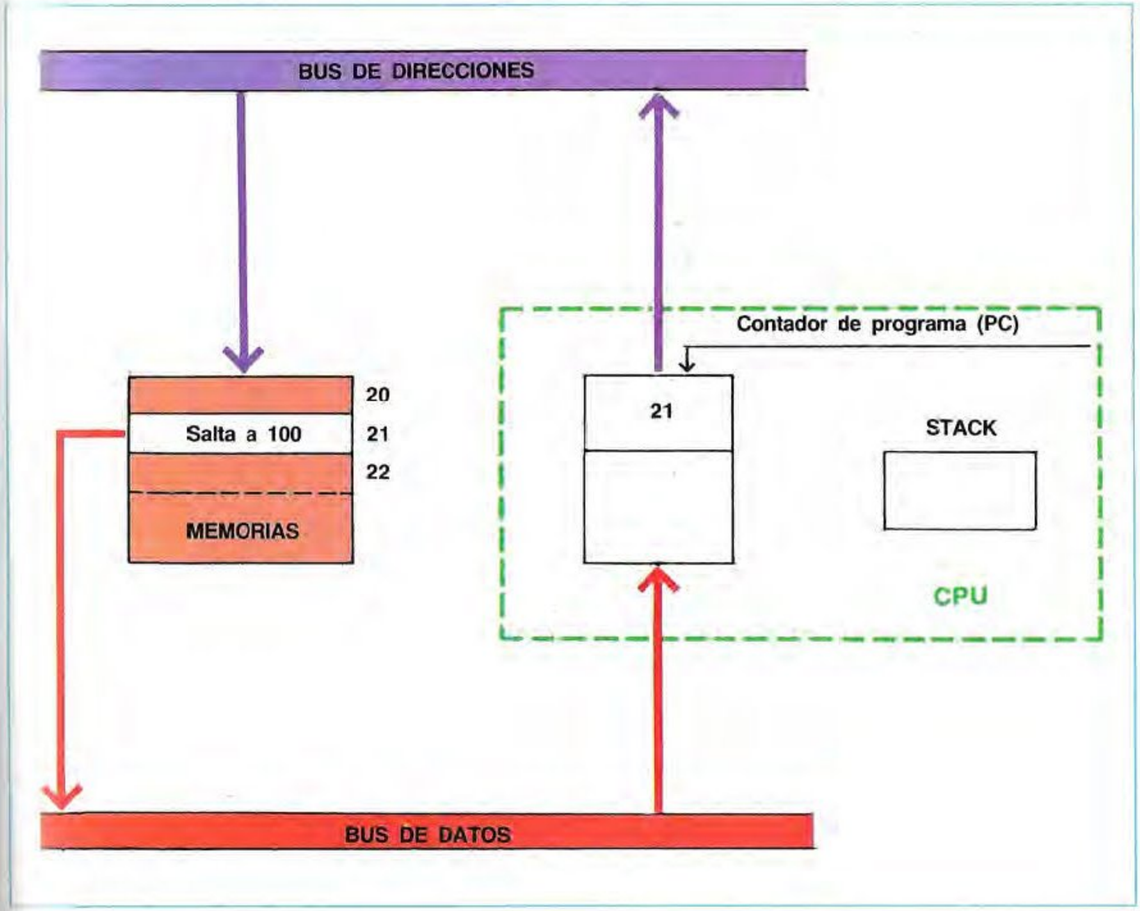
Al final de ese ciclo se suma 1 al contenido del PC; por lo tanto, en la instrucción siguiente se direccionará la memoria 1, y así sucesivamente. Con mucha frecuencia, en el desarrollo de un programa, no todas las instrucciones se han de ejecutar de forma secuencial. Puede darse el caso de que, tras una decisión lógica, haya que saltarse una parte del programa. Para memorizar el contenido del PC y luego poder retomarlo, al final del salto hay prevista una área de memoria denominada STACK (PILA).

Al llegar una instrucción de salto, la CPU salva el contenido del PC poniéndolo en el stack; en este momento el PC ya no contiene la dirección secuencial sino el nuevo valor. Por lo tanto, se toma y ejecuta la instrucción cuya dirección está contenida en el PC, es decir, se salta a la nueva dirección. Al final la CPU retoma el valor primitivo del PC y luego (tras incrementarlo) prosigue de forma secuencial hasta que se presenta una nueva instrucción de salto.

Fases de una instrucción de salto

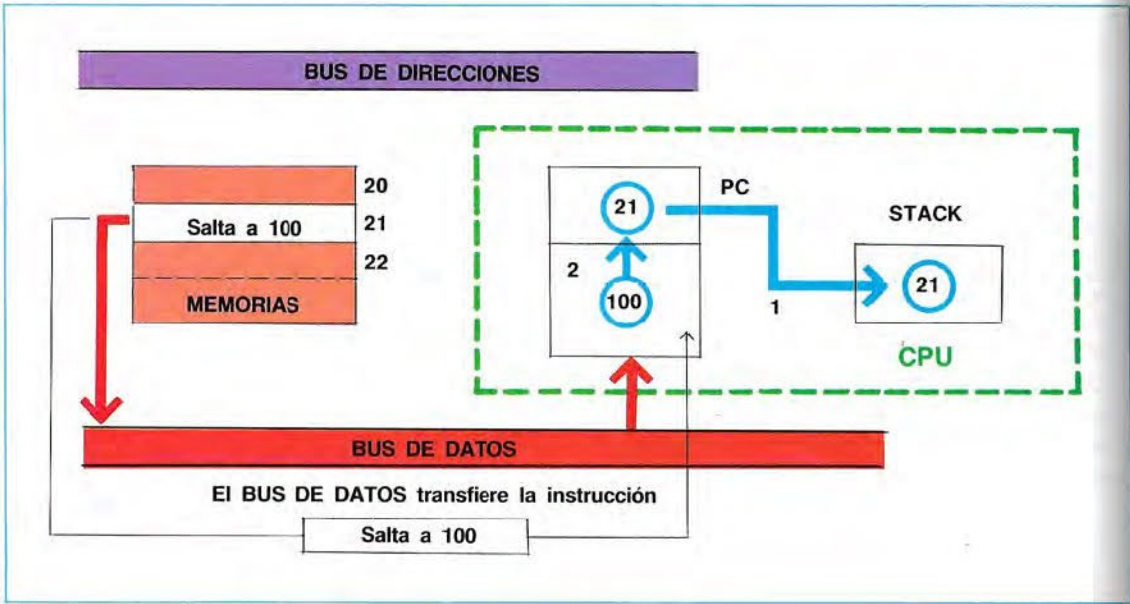
Primera fase

En la ejecución del programa, el PC ha llegado, por ejemplo, al valor 21. Este valor se introduce en el Bus de Direcciones. La memoria número 21 responde poniendo su contenido –por ejemplo, el código correspondiente a la instrucción «Salta a 100»– en el Bus de Datos. El stack todavía no interviene.

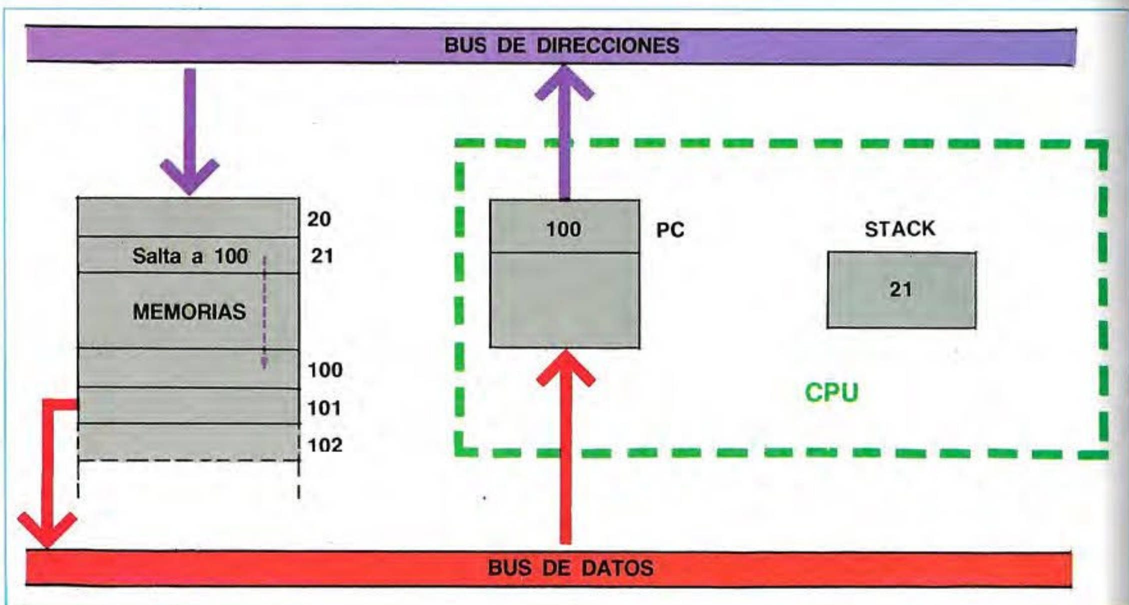


Segunda fase

La CPU reconoce la instrucción de salto; memoriza el contenido del PC en STACK **1** (o sea, memoriza cuál era la última instrucción antes del salto) y transfiere la nueva dirección (100) a PC **2**. Este valor (100) será la siguiente instrucción a ejecutar (ver el gráfico central).



En este momento (ver gráfico inferior) el proceso sigue desarrollándose normalmente: la CPU pone en el Bus de Direcciones el contenido del PC (100) y la memoria correspondiente responde poniendo su contenido en el Bus de Datos.



Esta nueva instrucción es ejecutada y el PC se incrementa en 1; por lo tanto, la CPU pasa a ejecutar la instrucción de la memoria 101, y así sucesivamente hasta encontrar una instrucción especial cuyo significado es «retorno» (RETURN).

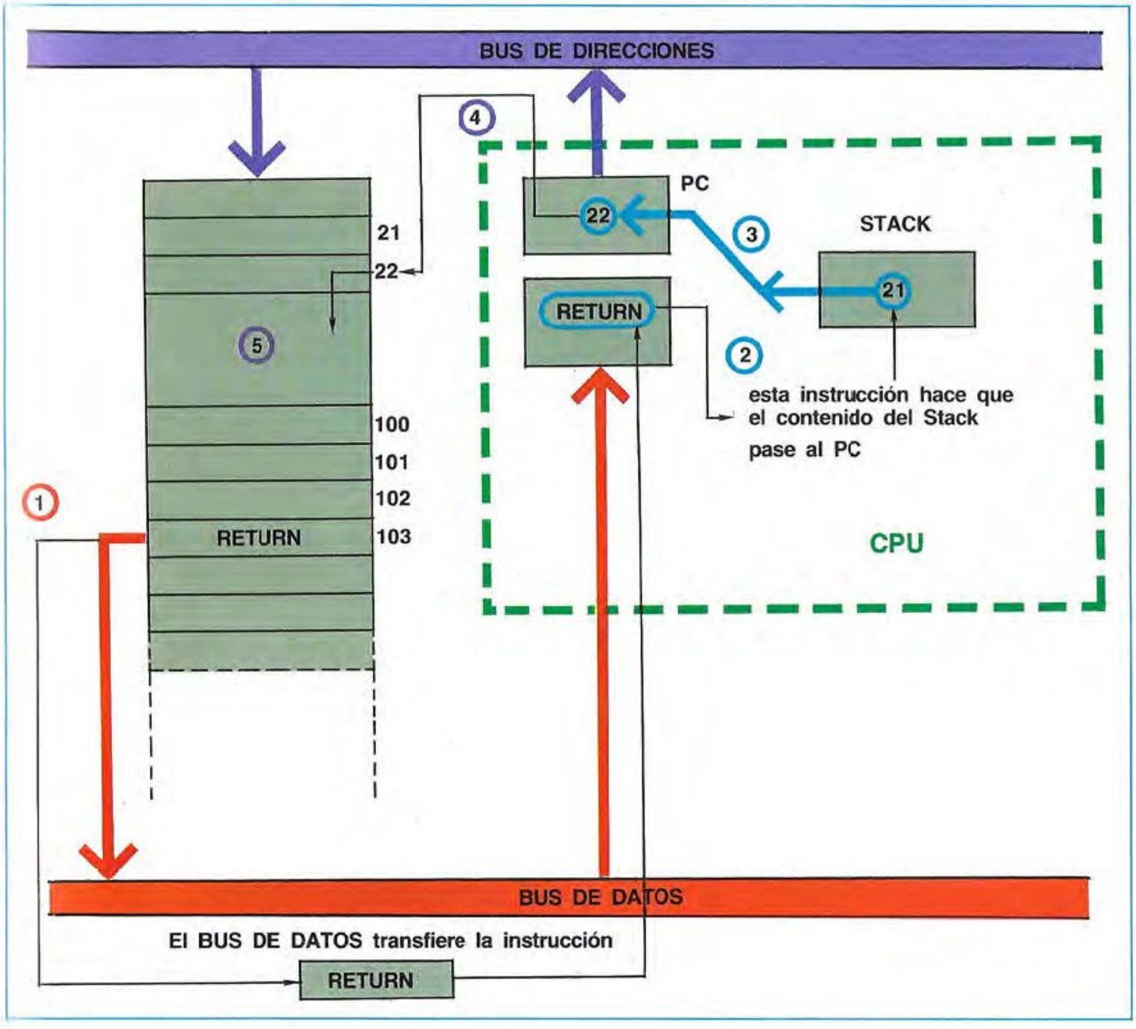
Cuando la CPU se encuentra con esta instrucción no hace más que tomar el contenido del stack y depositarlo, incrementado, en el PC.

En este momento el programa prosigue a partir de la instrucción siguiente al salto.

La CPU, tras el salto a 100 (ver gráfico siguiente), prosigue de forma secuencial con las instrucciones contenidas en 101,102,103. La memoria 103 contiene la instrucción RETURN. Al reconocer este código, la CPU toma el contenido del stack (en el ejemplo, 21), lo incrementa y lo escribe en el PC (en el gráfico, ①, ②, ③). A partir de aquí el programa vuelve a su flujo primitivo (instrucción 22 y siguientes: ④, ⑤). Esta técnica de recuperación de la dirección de retorno puede usarse para saltos sucesivos; por ejemplo, si la memoria 101 contuviera a su vez una instrucción de salto, se tendría el mismo mecanismo de funcionamiento, pero con dos recuperaciones de la dirección de retorno: la primera para el primer salto (21) y la segunda para el segundo (101). El primer retorno sería a 102 (101 + 1) y el segundo a 22 (21 +1).

En este caso el espacio ocupado por el stack es de dos células de memoria.

En algunos microprocesadores el stack está contenido en la CPU misma; esta solución tiene la ventaja de una mayor integración, pero limita la posibilidad de efectuar saltos sucesivos. En los sistemas de uso más general el stack puede estar en toda la memoria y, por tanto, los únicos límites son los de la propia memoria.

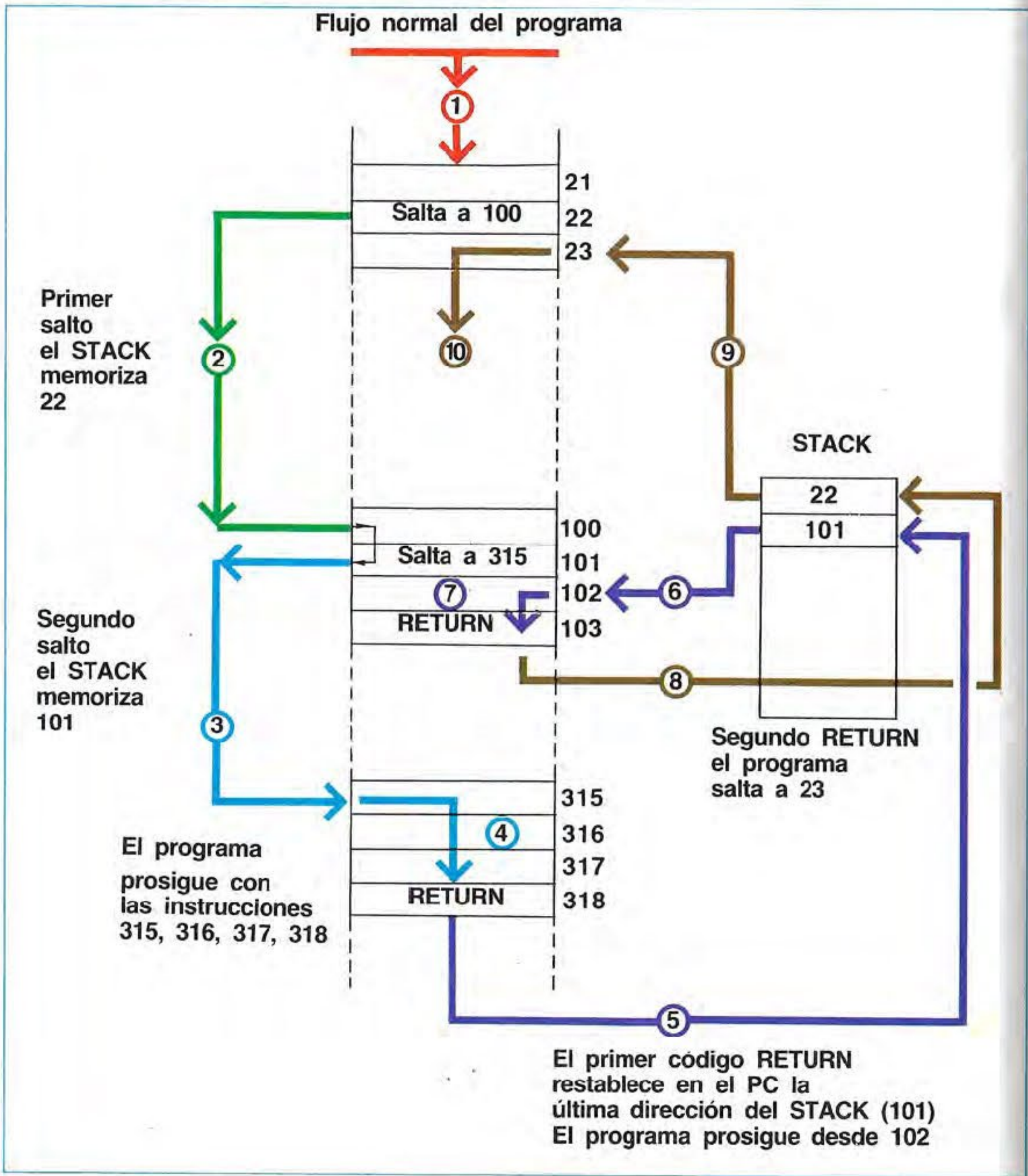


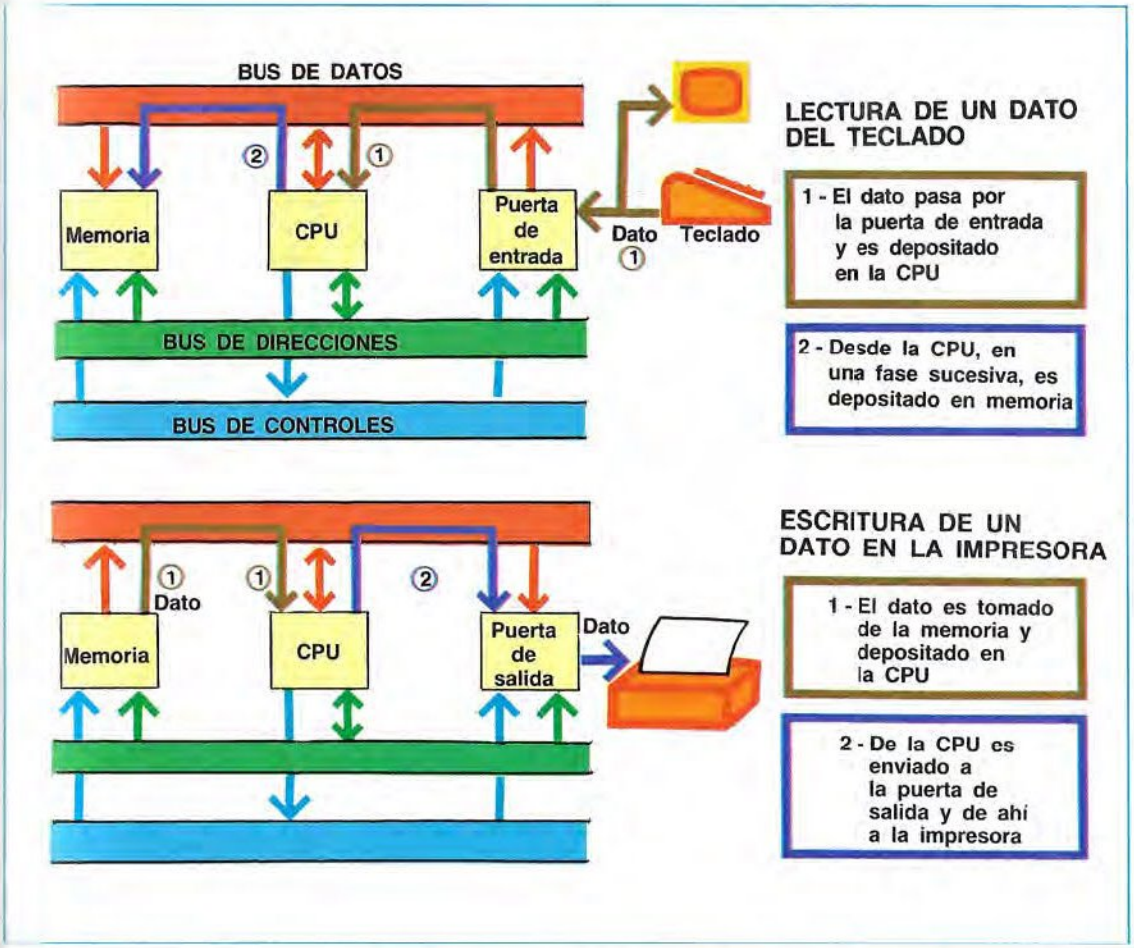
En el gráfico siguiente se esquematiza la lógica de funcionamiento con dos saltos sucesivos. La secuencia de las operaciones se indica con los números del **1** al **10.**

Interrupciones de programa

El desarrollo normal de un programa puede ser detenido momentáneamente por la aparición de una señal especial: INTERRUPT (Interrupción). Esta señal tiene por objeto interrumpir las operaciones en curso a fin de permitir la realización de otra función y, al término de la misma regresar al punto en que el programa se había interrumpido.

En todos los sistemas, los diversos periféricos (video, impresora, teclado) tienen distintos niveles de prioridad: normalmente, el periférico que ha de ser atendido en primer lugar es el teclado, que constituye el elemento mediante el cual se introducen las instrucciones; los demás periféricos pueden funcionar si el teclado no ha pedido el control.





Esta lógica de funcionamiento puede ser generada sólo mediante los Interrupt.

Las interrupciones de programa son necesarias para atender demandas de servicio procedentes del exterior y no previstas en el programa.

Un caso típico se da cuando, por cualquier motivo, se desea interrumpir un programa de impresión (por ejemplo, los datos están equivocados o la distribución no es correcta).

Un programa de impresión dedica todos los recursos del sistema a dicha operación; la CPU se ocupa de tomar los datos de las memorias, codificarlos adecuadamente y enviarlos a la impresora. En estas condiciones, el teclado, que es el órgano de mando, quedaría totalmente aislado, puesto que en el programa no hay previstas funciones de introducción de datos. No tendríamos manera de introducir órdenes sino al final del programa que se está llevando a cabo, es decir, cuando la CPU, liberada de la función de impresión, pudiera volver a comprobar si se habían tecleado nuevas instrucciones.

En realidad, el teclado envía un Interrupt, ante el cual la CPU interrumpe el programa y, tras reconocer al periférico que pide ser «escuchado», le pasa el control. El operador puede comunicarse con el sistema e introducir la orden de interrumpir la impresión.

Lectura de los datos

Los periféricos de un sistema se pueden dividir en dos tipos principales:

- **periféricos de entrada:** son los elementos a través de los cuales se introducen los datos en la máquina. En los microordenadores y ordenadores personales, para usos normales únicamente está prevista la utilización del teclado.
- **periféricos de salida:** son los elementos a través de los cuales la máquina se comunica con el operador. En los usos normales, son el video y la impresora.

Las operaciones de I/O (Entrada/Salida) se esquematizan en el gráfico superior.

Memorias que no olvidan

Incluso en el ordenador más potente, la unidad central de proceso (CPU, Central Processing Unit*) sólo puede efectuar una operación a la vez. Tras haber efectuado un cálculo memoriza su resultado, luego efectúa otro cálculo, memoriza su resultado en otro lugar, y así sucesivamente. Puede suceder que el ordenador tenga que efectuar miles de operaciones, todas en una fracción de segundo, antes de necesitar el resultado de la primera operación; pero cuando se requiera el primer resultado, tiene que estar disponible inmediatamente. Esto significa que la memoria ha de estar concebida de forma que los datos puedan introducirse y extraerse a voluntad, sin tener que buscarlos recorriendo los circuitos como si se pasaran las páginas de una agenda.*

Este tipo de memoria se denomina «de acceso aleatorio» (RAM, Random Access Memory*), y permite direccionar y catalogar exactamente toda información recibida o emitida.*

Como sabemos, el ordenador trata los datos que se introducen pasándolos previamente al sistema binario. Tras la codificación, un dato está constituido por una cadena de cifras binarias, y puede ser presentado a la memoria enviando, una a una, las cifras que lo componen, según lo que se denomina direccionamiento en serie (serial addressing).

Las cifras también pueden ser presentadas a la memoria todas a la vez mediante conductores en paralelo, cada uno de los cuales corresponde a una de las potencias de la base 2; este método se denomina direccionamiento en paralelo (parallel addressing).

La memoria está formada por filas y columnas de transistores ordenadas sobre una única plaqueta de silicio o chip*. La plaqueta ha de estar provista también de los circuitos necesarios para controlar, seleccionar, enviar, buscar e identificar los datos que entran o salen de la memoria.*

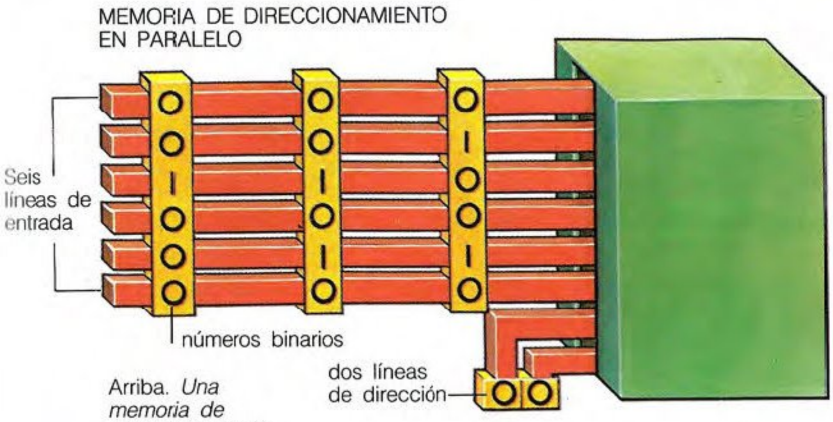
Hay dos tipos de memorias RAM de semiconductores: las estáticas *y las* dinámicas*. Una memoria RAM dinámica pierde su contenido tras un cierto tiempo, por lo que ha de ser «refrescada» por una señal de reloj* (clock) *periódica. Para esta operación hacen falta circuitos adicionales. Cada célula de memoria (o unidad de almacenamiento en el chip) es, sin embargo, más pequeña que una célula estática, que requiere al menos dos transistores para almacenar cada bit de información. La razón por la cual la memoria dinámica tiende a perder su contenido estriba en el hecho de que cada cifra 0 o 1 se memoriza almacenando una carga eléctrica (igual a cero o distinta de cero) en uno de los transistores de memoria. Éste se comporta como un simple condensador, y es bien sabido que cualquier condensador tiende a perder gradualmente su carga debido a los fenómenos de dispersión.*

En la memoria estática, por el contrario, la señal se almacena mediante un dispositivo denominado flip-flop *(o más exactamente* multivibrador biestable*), que, en esencia es un interruptor electrónico que sólo puede estar abierto* (on) *o cerrado* (off)*, dando la posibilidad de representar las dos cifras binarias 0 y 1 sin necesidad de almacenar cargas eléctricas.*

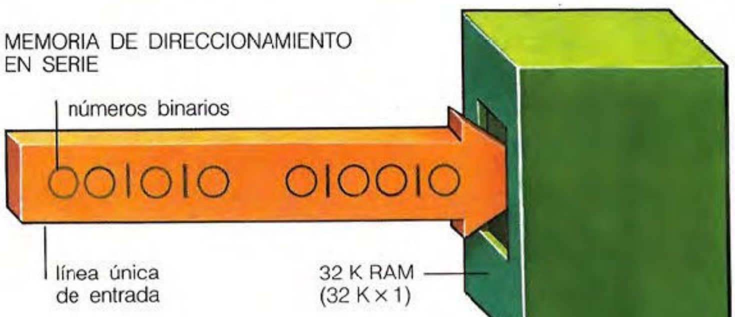
Los transistores más usados para la memorización son los del tipo MOSFET («transitor de metal-óxido-semiconductor de efecto de campo»). Miles de transistores de este tipo pueden imprimirse, gracias a técnicas de fotograbado especiales, en un centímetro cuadrado de silicio. Cada transistor está constituido por un microscópico cristal de material semiconductor, atravesado por una corriente eléctrica que fluye entre dos electrodos situados en los extremos, llamados fuente (source) *y* drenador (drain)*. Fundido con el semiconductor hay un tecer electrodo, que puede regular la corriente que fluye entre la fuente y el drenador, y que, de acuerdo con su función, se denomina* puerta (gate).

El transistor de efecto de campo que forma parte de una memoria, se utiliza como un interruptor que puede cerrarse enviando una señal a su puerta. En posición de circuito cerrado (apertura de la puerta) la corriente puede pasar de la fuente al drenador a través del dispositivo, cargando un condensador conectado con el transistor de efecto de campo. Este condensador puede así adoptar un estado de carga*, al que corresponde la cifra 1 en código binario, o un estado de* no-carga*, al que corresponde la cifra 0. Esta es la técnica que permite almacenar los datos en código binario.*

El conjunto formado por el MOSFET y el condensador se denomina célula de memoria*. La puerta de cada célula está conectada con una línea de selección que se activa con la orden correspondiente cuando se desea que una cifra*



Arriba. *Una memoria de direccionamiento en paralelo recibe las cifras binarias relativas a un dato simultáneamente, mediante varias líneas de entrada paralelas. La dirección en código, que indica dónde se ha de almacenar el dato, entra en el ordenador mediante dos líneas complementarias.*



Abajo. *En una memoria de direccionamiento serial todos los datos, incluidos los necesarios para determinar la dirección, entran por una línea única y son seleccionados por un circuito lógico.*

Ricky Blakely

sea cargada en el condensador de la célula. En una memoria de direccionamiento en serie, los datos entran por una única línea, y cada número binario va acompañado de una dirección de selección de columna de dos bits, que establece en cuál de las columnas ha de entrar el dato. Un circuito lógico se encarga de enviar la cifra a la dirección exacta. En una memoria de direccionamiento en paralelo, por el contrario, un número compuesto por cuatro bits es presentado por las cuatro líneas de los datos. Una señal 1 en la línea de dirección de columna almacenará el número en esa columna concreta. Mediante técnicas de este tipo, se puede memorizar un gran número de datos en un único chip.

Las memorias dinámicas de acceso aleatorio (RAM) han alcanzado una capacidad de 64.000 células (64K) por chip, en comparación con las 16K de las últimas memorias RAM estáticas.

En las memorias RAM estáticas de MOSFET normales, los resistores propiamente dichos casi no se utilizan. En lugar de cada resistencia de carga se utiliza un transistor con la puerta conectada al drenador, que constituye un resistor no lineal de reducida superficie.

Sin embargo, la firma Mostek ha vuelto a utilizar recientemente resistores propiamente dichos, empleando silicio polimorfo de implantación iónica, que, según se dice, ocuparía un poco más de espacio con respecto al sistema actual, pero requeriría un menor consumo de energía.

No todas las memorias de acceso aleatorio utilizan la tecnología MOS (metal-óxido-semiconductor). También se usan memorias denominadas bipolares, que se suelen realizar con parejas de flip-flop, a la manera de las memorias estáticas MOS, con el consiguiente aumento de la superficie necesaria para cada célula. Las memorias bipolares pueden ser mucho más veloces que las memorias dinámicas, aunque las memorias dinámicas MOS son las más económicas. Una variante de la memoria estática de acceso aleatorio MOS es la MOS complementaria, o CMOS. En ella, los resistores de carga de la pareja cruzada de transistores de un flip-flop están constituidos por transistores polarizados en dirección contraria con respecto a la de los transistores de conmutación. La resistencia ofrecida por los transistores de carga es teóricamente infinita, por lo que no pasa co-

rriente y no se disipa energía. El resultado es que las memorias RAM estáticas CMOS consumen menos energía que los otros tipos, y esto explica por qué, a pesar de su funcionamiento más lento, son adoptadas en las actividades espaciales. Si bien en las memorias CMOS teóricamente no pasa por el transistor corriente alguna, en la práctica se produce un flujo de corriente durante el breve período en que el flip-flop *se conmuta, pasando de un estado a otro. Este inconveniente, provocado por los efectos de las capacidades parásitas, ha sido eliminado recientemente por la RCA, que ha puesto a punto un método para la realización de memorias CMOS sobre un sustrato de zafiro. Esta técnica cuadruplica la velocidad de funcionamiento, pero el zafiro es mucho más caro que el silicio, por lo que, si bien las investigaciones aún están en marcha, el futuro de esta técnica se presenta un tanto incierto.*

Una desventaja que presentan todas las memorias RAM de semiconductores es la de ser volátiles, pero se han estudiado numerosos sistemas para subsanar este defecto. Si se desconecta la alimentación de la memoria, por ejemplo, se puede efectuar la transferencia inmediata de los datos más importantes a un almacenamiento magnético de algún tipo (disco, cinta).

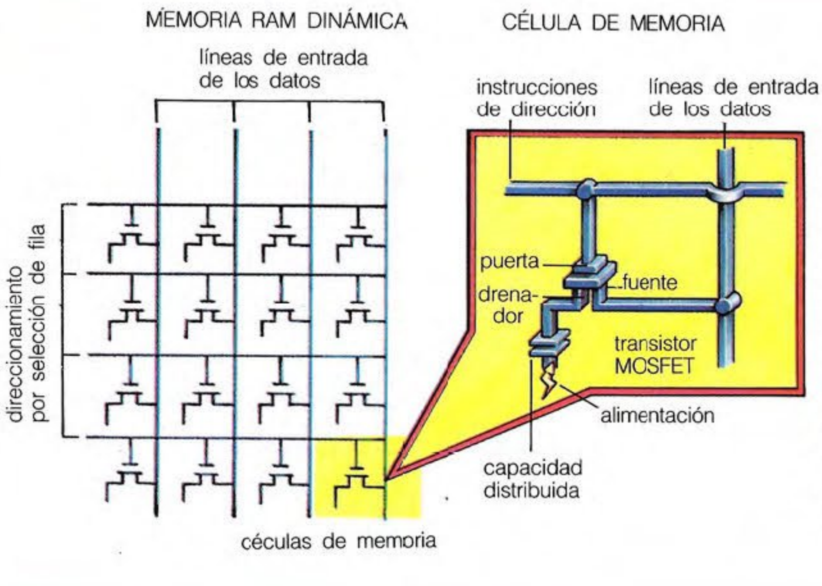
Hay también memorias de semiconductores no volátiles, que se utilizan para almacenar programas por tiempo indefinido, como los de un videojuego o los necesarios para dirigir las operaciones internas de un ordenador, o para instruirlo sobre la forma de desarrollar un proceso matemático. Estas memorias sólo pueden suministrar un determinado conjunto de datos, que son continuamente leídos y utilizados por el ordenador, por lo que se denominan memorias sólo de lectura, *o ROM* (Read Only Memory) *en la jerga de la industria de los ordenadores. Las memorias ROM están formadas por una simple matriz de diodos, y su programación se efectúa de una vez por todas, fundiendo mediante la aplicación de alta tensión los diodos a descartar.*

La falta o presencia de conexiones entre las líneas puede servir para indicar un estado lógico 1 o 0. La sencillez de una memoria ROM explica su bajo costo y la elevada densidad con que los datos pueden ser memorizados en ella.

Las primeras memorias ROM ya no podían modificarse tras haber sido programadas, pero pronto aparecieron componentes que podían «limpiarse» exponiendo la matriz de diodos a la luz ultravioleta a través de una ventanilla abierta en la parte superior del dispositivo.

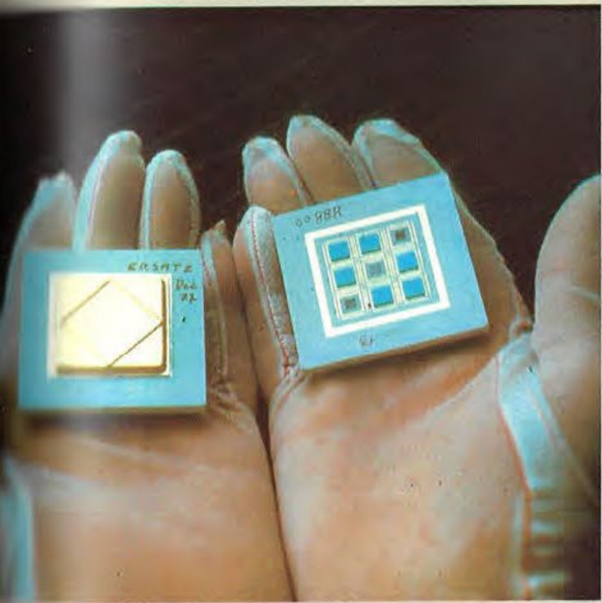
Las memorias de este tipo se designaron con la sigla EPROM (Erasable Programmable Read Only Memory), en contraposición a las memorias ROM programables, denominadas PROM.

Las memorias EPROM han alcanzado ya capacidades de 64.000 bits, y los trabajos para realizar versiones de 128.000 y 256.000 bits se ha-



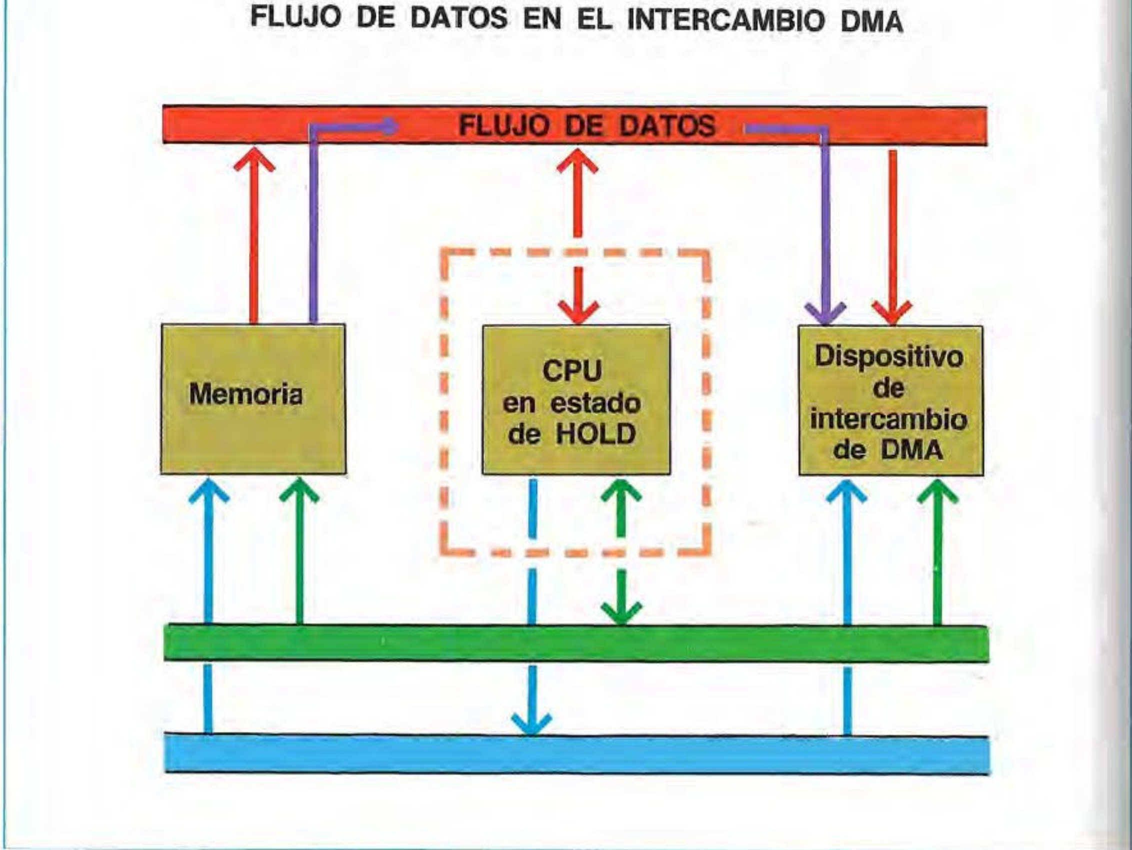
Estructura de una memoria dinámina de acceso aleatorio, o RAM. Este dispositivo contiene miles de transistores de metal-óxido-semiconductor con efecto de campo, o MOSFET, cada uno de los cuales, junto con una capacidad distribuida, constituye una célula de memoria. El MOSFET funciona como un conmutador que permite a las capacidaes cargarse con los datos que llegan.

Ricky Blakely



Dos fases del montaje de los chips de memoria sobre soportes de cerámica.
A la derecha, los chips de soporte; a la izquierda, los chips aislados y recubiertos. En este momento, el módulo está listo para ser montado.

llan actualmente en estado avanzado.
Otra variante de la memoria ROM es la eléctricamente modificable, o EAROM (Electrically Alterable Read Only Memory), *con la que no es necesario borrar toda la memoria para reprogramar una parte, puesto que la simple aplicación de tensiones más altas de lo normal provoca la reprogramación de las células por separado.*
Hay también otros prometedores sistemas para almacenar datos numéricos, que todavía no se han difundido comercialmente. Entre los ejemplos más importantes están las memorias de burbujas magnéticas y las basadas en dispositivos de corrimiento de carga (CCD: Charge-Coupled Devices*).*
En el caso de las burbujas magnéticas, los datos se memorizan aprovechando las propiedades de ciertos cristales que, bajo la influencia de un campo magnético, almacenan lo que podríamos llamar burbujas dotadas de polaridad magnética. La polaridad de las burbujas determina si se ha almacenado un 1 o un 0.
Otra posibilidad es la de generar o no una burbuja, con el significado de 1 o 0 respectivamente. Estos dispositivos están capacitados para almacenar un grandísimo número de datos, los cuales, sin embargo, y para que el dispositivo funcione correctamente, han de ser memorizados en el equivalente de una bobina magnética cerrada sobre sí misma. Por consiguiente, se necesita un elemento especial para «escribir» la información en el soporte, más o menos como en el caso de la cabeza de una grabadora, y otro elemento para «leer» la información. El lector, siendo fijo, puede leer la información sólo cuando ésta se vuelve cíclicamente accesible debido al movimiento del soporte.
Todo esto hace que el tiempo de acceso sea bastante largo, pero, como en todos los dispositivos magnéticos, las burbujas tienen la ventaja de no ser volátiles; además, la densidad de concentración permitida por este sistema es del orden de diez veces mayor que la de una memoria de semiconductores.
Las memorias de este tipo están empezando a ser utilizadas por empresas norteamericanas como la IBM, así como por algunas de las firmas japonesas que se dedican a la fabricación de ordenadores.
Las memorias por corrimiento de carga utilizan una técnica basada en los semiconductores, que las convierte en algo parecido a larguísimos archivos corredizos. Los datos se almacenan en larga sucesión, y se puede acceder a ellos, como en la memoria de burbujas, sólo cuando pasan frente a cabezas lectoras especiales. Una memoria CCD es, desde el punto de vista físico, un transistor con efecto de campo de gran longitud, dotado de un cierto número de salidas (o electrodos) diseminadas entre los dos extremos, casi como una manguera para regar el jardín a lo largo de la cual se hubiera practicado un gran número de agujeros. Pero la cantidad de carga que fluye a lo largo del transistor puede variarse de forma más precisa que el agua de la manguera, y a cada electrodo se presenta un cuadro completo de la serie de señales que llegan a la entrada del dispositivo. Para lograr que los datos avancen a lo largo del dispositivo de corrimiento de carga, se hace necesario suministrar con regularidad una señal de reloj, y el tiempo de acceso a cada bit de información es mayor que en otros dispositivos equivalentes.
En la actualidad se cree que este sistema resultará poco costoso, pero queda por demostrar que sea adecuado para la utilización generalizada.



En la lectura, por ejemplo, desde el teclado, el dato es transferido primero a la puerta de entrada, luego a la CPU y finalmente a la memoria.
Por el contrario, en la salida, el dato es tomado de la memoria, transferido a la CPU y luego enviado a la puerta de salida.
Esta gestión de las operaciones I/O, que conlleva el paso a través de la CPU, no puede efectuarse a gran velocidad y es, por tanto, adecuada para la transferencia de datos hacia periféricos lentos (como el teclado y la mayor parte de las impresoras). Para intercambios de datos más rápidos se utiliza la técnica DMA (Direct Memory Access), que consiste en un intercambio directo entre la memoria y los aparatos que reciben o transmiten los datos. En este intercambio no interviene la CPU, por lo que su actividad queda suspendida hasta el final del intercambio. Este estado de la CPU en actividad suspendida se denomina HOLD.
En el gráfico superior se muestra el esquema de un intercambio DMA.

Normalmente, este tipo de intercambio se adopta para las Unidades de Disco, puesto que son dispositivos que han de intercambiar grandes cantidades de datos.
Un disco pequeño puede contener entre 250.000 y 1.200.000 caracteres, y en otros tipos se puede llegar a los 5-10 millones de caracteres; por lo tanto, un mínimo ahorro de tiempo en cada carácter se traduce, globalmente, en un tiempo considerable.
Por otra parte, en los programas los datos son continuamente tomados de la unidad de disco, elaborados y luego vueltos a depositar en ellas; este constante acceso al disco hace indispensable el uso del DMA.
La elección entre uno u otro tipo de intercambio es automática, puesto que ciertos programas especiales se encargan de la selección y ejecución sin que el usuario ni el programador tengan que preocuparse de ello. Este tipo de programas forma parte del Sistema Operativo que se suministra junto con la máquina.

Memorias

Las memorias son los dispositivos en los que se almacenan los programas y los datos que dichos programas utilizan o producen. Se trata de una memorización provisional, puesto que al escribir un nuevo programa se destruye el anterior; igualmente, al desconectar la máquina se pierde el contenido de las memorias (excepto en algunos tipos especiales). La memorización permanente tiene lugar en disco, por lo que todos los programas y datos que hay que volver a utilizar han de ser transferidos a un disco (o a una cinta magnética).

El Área de Memoria, es decir, la cantidad de memoria que posee una máquina, se divide en cuatro bloques fundamentales:

1 / área reservada a los Interrupt
2 / área de sistema
3 / área de los programas de servicio
4 / área del usuario

En el área de los Interrupt se memorizan las instrucciones que permiten el funcionamiento de los Interrupt (interrupciones del programa). El área de sistema y la de los programas de servicio están dedicadas al Sistema Operativo, es decir, al conjunto de programas que realizan las funciones básicas, tales como la impresión, lectura del teclado, gestión de los discos, etc. La parte restante es el área que está a disposición de los programas de usuario y sus datos correspondientes.

Por término medio, los microordenadores y ordenadores personales tienen entre 32.000 y 64.000 caracteres de memoria. De estos, unos 28.000 a 58.000 están a disposición, según la complejidad del Sistema Operativo. En los microordenadores más sofisticados y en los miniordenadores se superan los 170.000 a 250.000 caracteres, hasta llegar a 2-4 millones en las máquinas de más envergadura. En el gráfico se enumeran los distintos tipos de memoria.

Memorias de lectura y escritura: se designan con la sigla RAM. En ellas se puede tanto leer como escribir. Se dividen en:

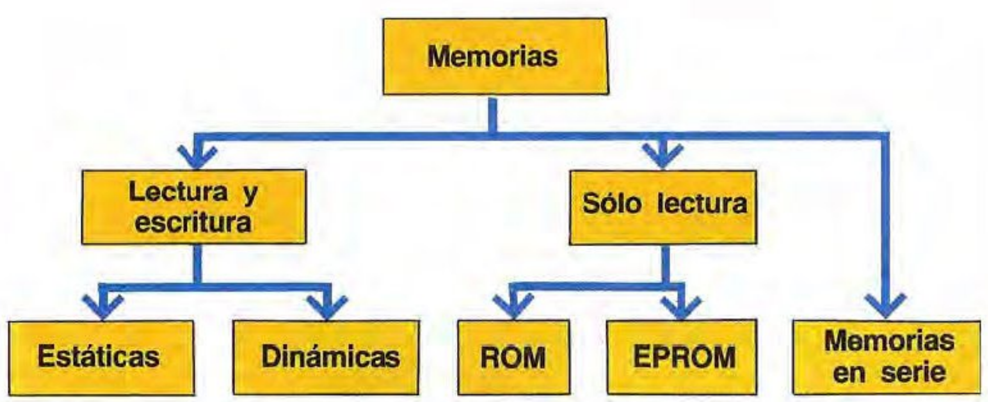
- **estáticas**
- **dinámicas**

Las de tipo estático, una vez escritas, no necesitan más; el dato puede ser tomado incluso después de un cierto tiempo. En las del tipo dinámico, por el contrario, hay que reescribir el dato a intervalos regulares (varios centenares de veces por segundo), pues de lo contrario se borra. La operación de reescritura de las memorias dinámicas se denomina REFRESH (REFRESCO).

Memorias sólo de lectura: Sólo pueden ser leídas. Se usan para programas que no se han de modificar. Son de dos tipos:

- **ROM**
- **EPROM**

El tipo ROM (Read Only Memory) se fabrica con los datos ya escritos y, por lo tanto, no se puede modificar. El tipo EPROM puede ser borrado con rayos ultravioleta y luego reprogramado.





Archivo Curcio/Buffetti Data

Instrumental para la programación de memorias EPROM.

Memorias en serie: son memorias de utilización muy especial, que permiten grandes capacidades pero necesitan circuitos complejos. Se utilizan con poca frecuencia.

Periféricos

Video (monitor): es muy similar a un televisor normal, hasta el punto de que en algunos ordenadores personales económicos se prevé el uso de un televisor en su lugar (mediante un circuito auxiliar). Los monitores se dividen en dos tipos principales:

- **monitores alfanuméricos**
- **monitores gráficos (también denominados videográficos)**

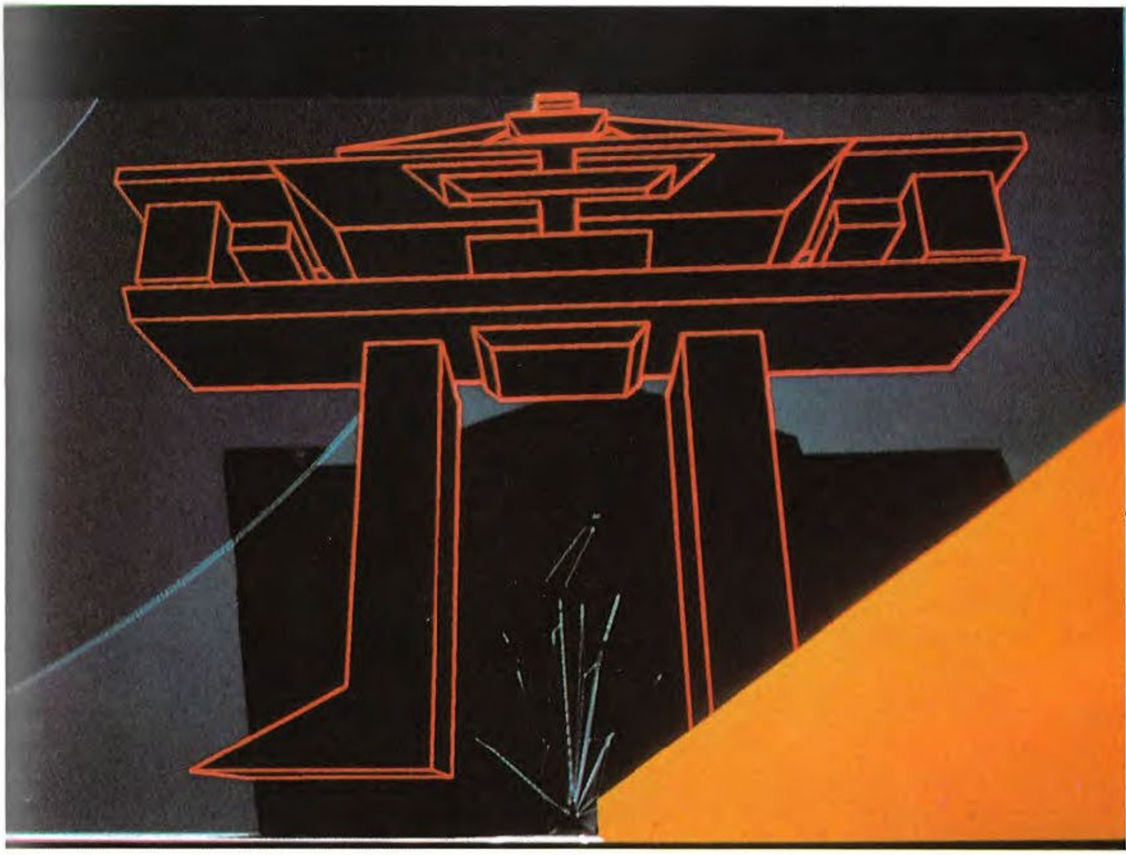
En el tipo alfanumérico sólo se pueden escribir letras y números; en el tipo gráfico se pueden realizar también dibujos.
La diferencia principal estriba en el número de puntos que componen la pantalla; cuanto mayor es dicho número, mejores posibilidades gráficas tiene la pantalla.
El tipo videográfico existe también en la versión en color. Las principales características de una pantalla de video son:

- **dimensiones**
- **número de líneas y columnas** (para el tipo alfanumérico)
- **color**

Las dimensiones se indican siempre en pulgadas. El número de líneas y columnas que la pantalla puede contener depende del tamaño de los caracteres; algunos valores típicos son:

16 líneas × 64 columnas
25 líneas × 80 columnas
40 líneas × 132 columnas

Algunos monitores tienen una pantalla que se utiliza con distintos formatos; el programador puede elegir el más adecuado en función de la cantidad de datos que hay que visualizar.
En los videográficos, en vez de las líneas y columnas se indican los elementos de figura, es decir, en cuántos puntos se divide (vertical y horizontalmente) la pantalla; un valor típico es 512 × 256 puntos gráficos.



1982 © Walt Disney Productions



1982 © Walt Disney Productions

Utilización del grafismo computerizado en el cine de animación. Dos imágenes de la película "Tron".

Teclado

Normalmente está dividido en dos secciones: la parte alfanumérica, que incluye tanto los caracteres alfabéticos como los números, y la parte numérica, que contiene sólo los números. En algunos tipos de máquina cada tecla tiene un doble significado: el primero es el carácter alfanumérico normal, y el segundo (que se activa con un código especial) es una instrucción completa. El empleo de este tipo de teclado permite un notable ahorro de tiempo, puesto que, en la escritura de un programa, muchas de las instrucciones están ya preparadas y se introducen en la máquina pulsando una sola tecla.

Impresora

En los microordenadores, los tipos más usados son las impresoras:

- **de aguja**
- **de margarita**
- **paralelas**

Impresoras de aguja: este tipo toma su nombre de la manera en que se forma un carácter.
La cabeza impresora está constituida por un cierto número de agujas (movidas por electroimán) que avanzan según la letra o el símbolo a imprimir. El carácter resultante no está escrito con un trazo continuo, sino que está formado por un conjunto de puntos muy próximos.
En la tabla inferior se muestran algunos tipos de caracteres que se pueden obtener con impresoras de aguja. Las ventajas de estas impresoras son: bajo costo, velocidad de impresión mediana, posibilidad de imprimir distintos tipos de caracteres. Su mayor desventaja es la mediocre estética de los caracteres.

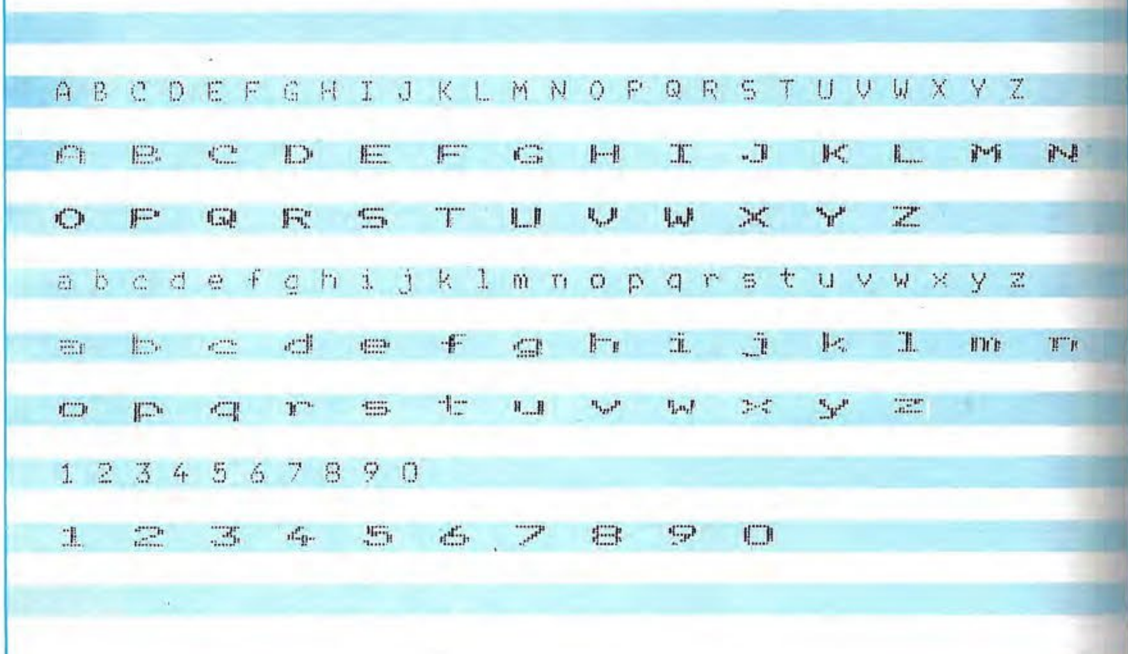
Impresoras de margarita: en este tipo cada carácter está ya listo y sólo hay que imprimirlo sobre el papel. Las margaritas son idénticas a las utilizadas en las maquinas de escribir. Estas impresoras tienen costos más altos que las anteriores, velocidades de impresión más bajas (cerca de la mitad) y el cambio de caracteres sólo puede hacerse sustituyendo la margarita.

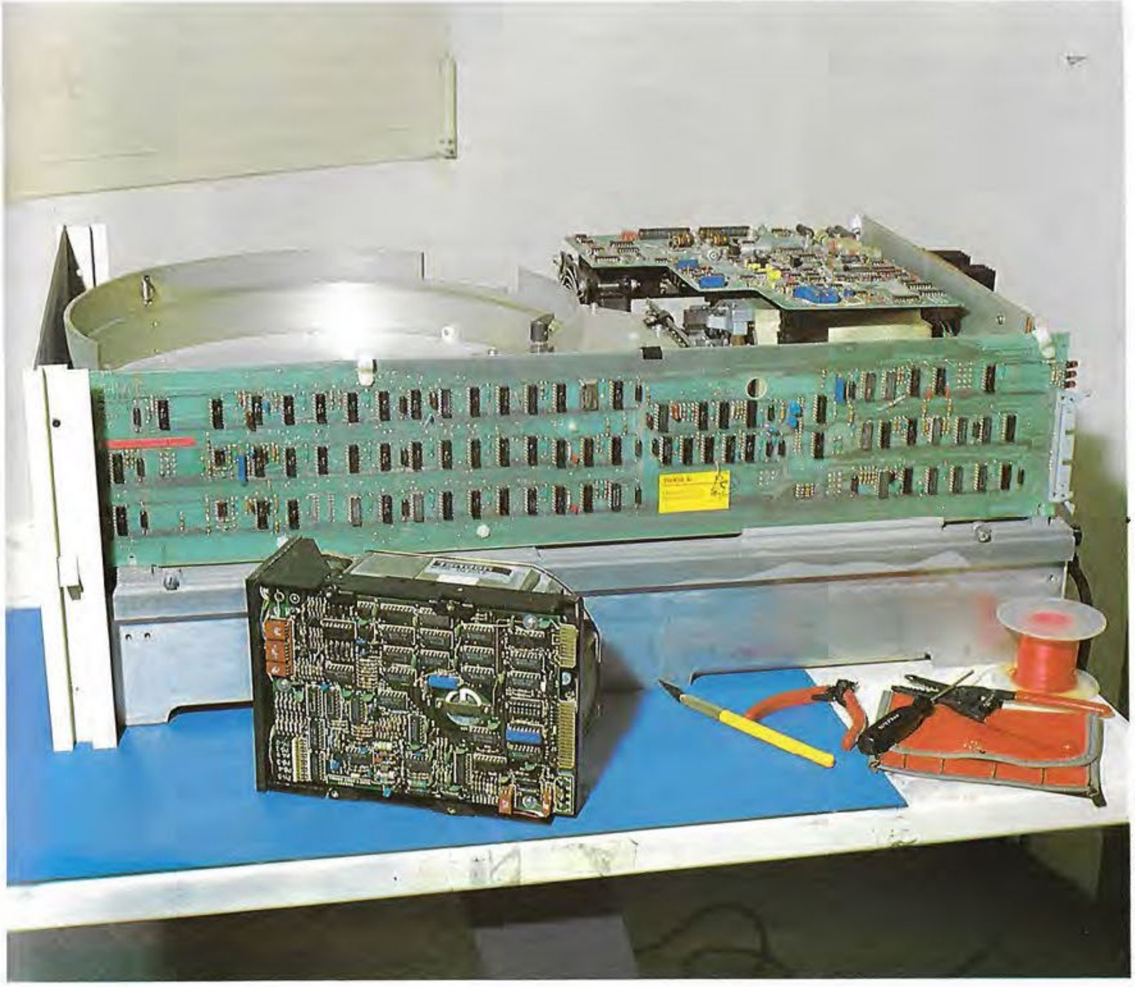
Impresoras paralelas: constituyen el tipo más rápido; pueden imprimir hasta unas 600 líneas por minuto. Su costo es muy elevado, por lo que su empleo sólo se justifica en los miniordenadores de mayor tamaño.

Unidades de disco

En los ordenadores personales y microordenadores se utilizan dos tipos: de disco flexible y de disco fijo.

CARACTERES DE IMPRESORAS DE AGUJAS





Archivo Curcio/Buffetti Data

Comparación entre dos tipos de unidades disco. A pesar de la notable diferencia de tamaño, la capacidad de memorización es la misma en ambas unidades.

El primer tipo de disco (floppy) es amovible y tiene una capacidad de memorización comprendida entre 250.000 y 1.200.000 caracteres (según las dimensiones y el tipo).
El segundo tipo de disco, que normalmente es inamovible, tiene una capacidad mínima de 5 a 10 millones de caracteres.

Los floppy pueden ser:

- **de una o dos caras**
- **de simple o doble densidad**

La primera característica indica si el disco puede ser utilizado por una o ambas caras; la segunda indica la mayor o menor densidad con que pueden escribirse los datos.
Naturalmente, el tipo de una cara y densidad simple es el que puede contener menor cantidad de datos; viceversa, el tipo de dos caras y doble densidad es el que posee la mayor capacidad de memorización. En los sistemas más grandes (miniordenadores), además de los tipos que hemos visto, se utilizan discos rígidos fijos y discos rígidos móviles.
Los de tipo fijo poseen grandes capacidades de memorización y su costo es elevado; por lo tanto, sólo se utilizan en aplicaciones de cierta complejidad. Los del segundo tipo (rígido móvil) tienen capacidades del orden de 10.000.000 de bytes (caracteres) y un costo más bajo.
En la fotografía superior se comparan dos unidades de disco, ambas con una capacidad de 10 Mbytes (10.000.000 de caracteres); la más grande es de disco rígido móvil y la otra es una unidad de floppy múltiple. La fotografía muestra claramente la gran diferencia de tamaño.

Funcionamiento de las memorias sólo de lectura

Las memorias sólo de lectura (ROM), como ya sabemos, sirven para guardar algoritmos permanentes especiales cuya finalidad es ordenar la ejecución de determinadas acciones. El denominado software permanente, *más conocido como* firmware, *se utiliza para realizar una amplia gama de funciones repetitivas en muchos aparatos, y el ejemplo siguiente describe el empleo de una memoria ROM para controlar un display de siete segmentos, destinado, por ejemplo, a presentar números en el visor de una calculadora de bolsillo.*

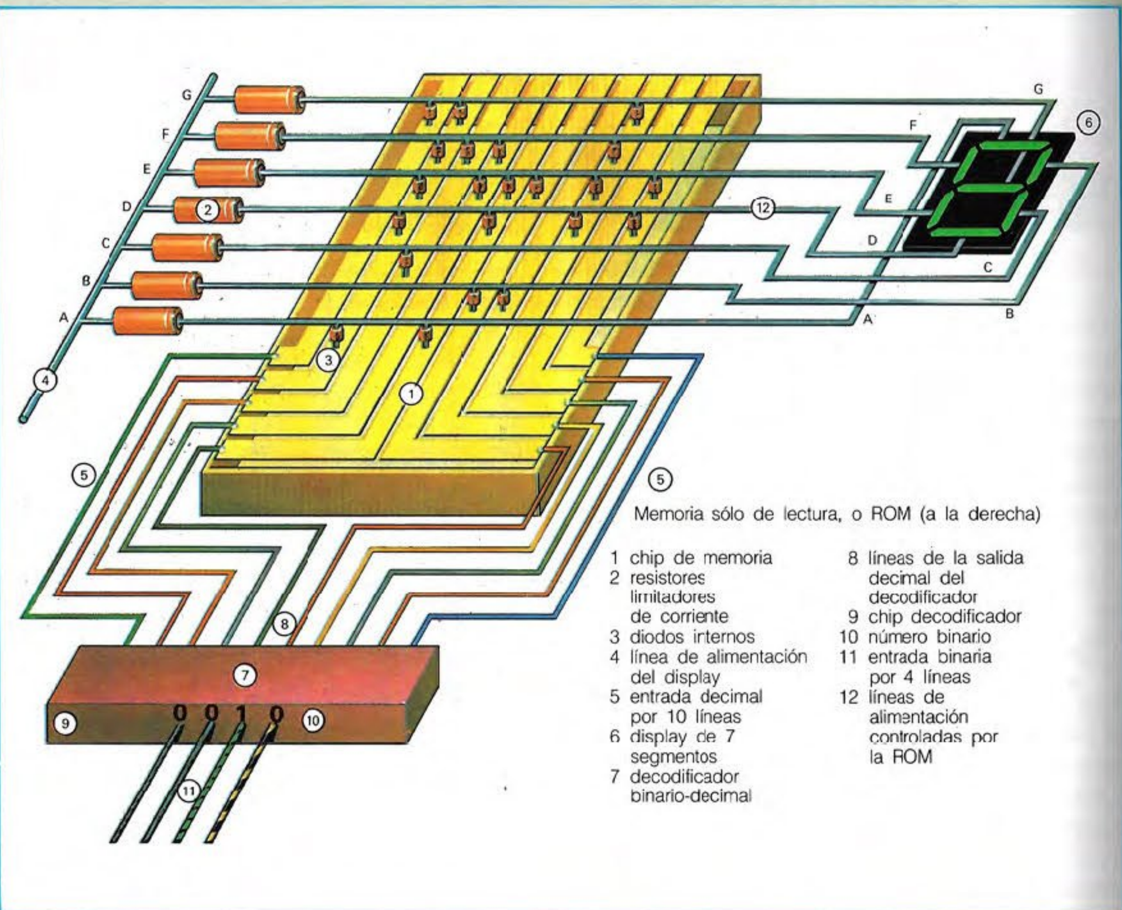
Como ya sabemos, las señales procedentes del microprocesador o de la memoria son codificadas en la simbología binaria. Un display numérico de siete segmentos es un dispositivo digital en base 10, y como tal ha de tener la posibilidad de recibir en entrada diez señales diferentes.

El problema de la conversión se resuelve con el empleo de un decodificador binario-decimal, compuesto por un conjunto de puertas lógicas. Las cuatro líneas de entrada examinadas en este ejemplo pueden llevar al decodificador las cifras binarias que representan los números del 0 al 9; para representar la cifra decimal comprendida entre 0 y 9 en que han de convertirse los datos binarios, hacen falta, sin embargo, diez líneas de salida.

En ausencia de señales binarias en la entrada, las diez líneas de salida están permanentemente sometidas a una tensión continua, la misma a la que opera el display (que normalmente es a 5 voltios).

Cuando, por ejemplo, el decodificador recibe el número binario 0000 (0 en el sistema decimal), da en la salida un 0, lo que se logra produciendo una falta de tensión en la primera línea de salida decimal (la del 0) y un 1 (presencia de tensión) en las demás líneas (cifras de 1 a 9).

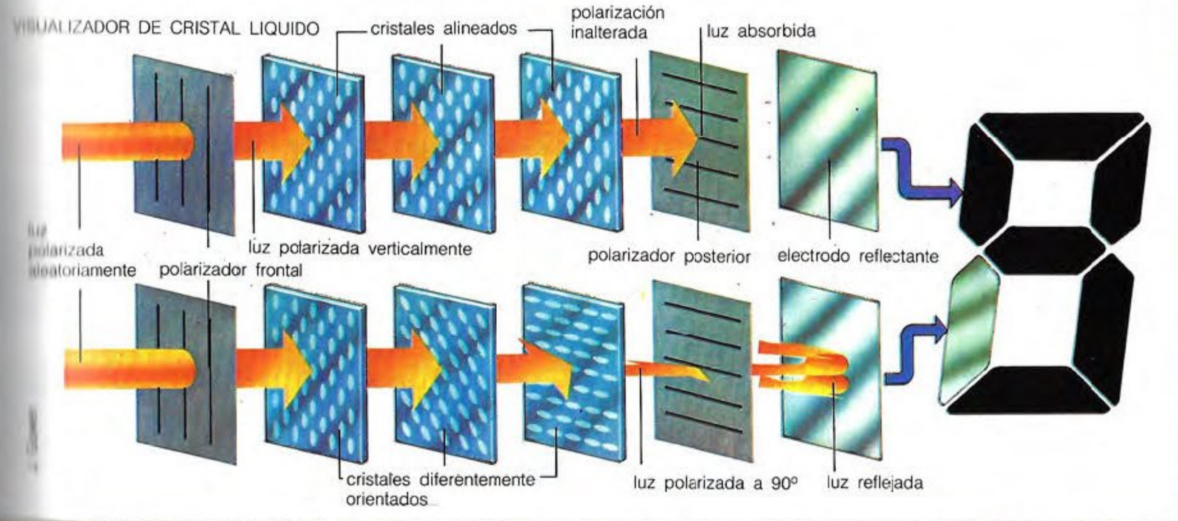
Si el decodificador recibe el número binario 0001 (1 en el sistema decimal), en tal caso saldrá un 0 (ausencia de tensión) en la segunda línea (que representa el decimal 1), mientras



Memoria sólo de lectura, o ROM (a la derecha)

1 chip de memoria
2 resistores limitadores de corriente
3 diodos internos
4 línea de alimentación del display
5 entrada decimal por 10 líneas
6 display de 7 segmentos
7 decodificador binario-decimal
8 líneas de la salida decimal del decodificador
9 chip decodificador
10 número binario
11 entrada binaria por 4 líneas
12 líneas de alimentación controladas por la ROM

Willy Stephen



que las demás líneas llevarán un 1 (presencia de tensión). El procedimiento se puede repetir para todas las demás cifras decimales, y una determinada cifra será indicada en la salida del decodificador por una falta de tensión en la línea correspondiente.

Hay una razón muy simple para usar la falta de tensión como indicador, igual que en las lógicas invertidas: si en una línea no hay tensión, significa que está unida a tierra y que tiene, por tanto, la propiedad de anular la tensión en cualquier línea que se ponga en contacto con ella. Esto permite dejar sin alimentación uno de los segmentos del display para que no se ilumine. La exclusión selectiva de los segmentos permite obtener sobre un display de siete segmentos las cifras de 0 a 9.

El cometido de la memoria ROM es, simplemente, el de hacer que se eliminen los adecuados segmentos del display para cada emisión del decodificador. Por ejemplo, con un 0 en la primera línea de salida del decodificador, que representa el decimal 0, hay que desactivar el segmento horizontal del centro del display, tras lo cual aparecerá el 0.

La realización de una memoria ROM muy sencilla, capaz de llevar a cabo las funciones necesarias, se articula en varias fases sucesivas. En primer lugar, se compila una tabla que indique qué segmentos del display han de iluminarse para cada combinación de cuatro bits que puede presentarse en la entrada. Luego se realizan las conexiones necesarias en una matriz de diodos, de la forma que se indica en la figura. Si un diodo está entre la primera línea de salida del decodificador y la línea de alimentación G, cada vez que en la primera se produce una falta de tensión, automáticamente queda sin alimentación el segmento central del display. Los diodos que conectan las líneas de alimentación de A a F con la línea 0 del decodificador no sirven; por lo tanto, son fundidos deliberadamente y eliminados de la matriz durante el proceso de programación de la ROM, sometiéndolos a una tensión muy alta. Así, gracias al sencillo método de la matriz de diodos, la memoria ROM puede ser programada para realizar por largo tiempo una determinada función.

Como hemos visto, un display numérico (o visualizador digital) es gobernado por una unidad de salida (interface) que recibe del ordenador central un dato a mostrar, lo decodifica y suministra en salida los impulsos necesarios para accionar el display. El visualizador de diodos de emisión luminosa (LED) puede mostrar las cifras de 0 a 9 activando las adecuadas combinaciones de los siete segmentos que lo componen.

Utilizando cristales líquidos, también se necesitan siete segmentos para formar la cifra 8, mientras que activando los segmentos oportunos se pueden formar las demás cifras de 0 a 9.

Estos visualizadores se basan en las propiedades de los cristales nemáticos dispersos en un líquido interpuesto entre una lámina frontal transparente y otra posterior muy reflectante. Aplicando una diferencia de potencial entre los segmentos de la lámina frontal y los de la posterior, el líquido se vuelve opaco y entonces, al alterar las propiedades de reflexión, produce una barrita visible.

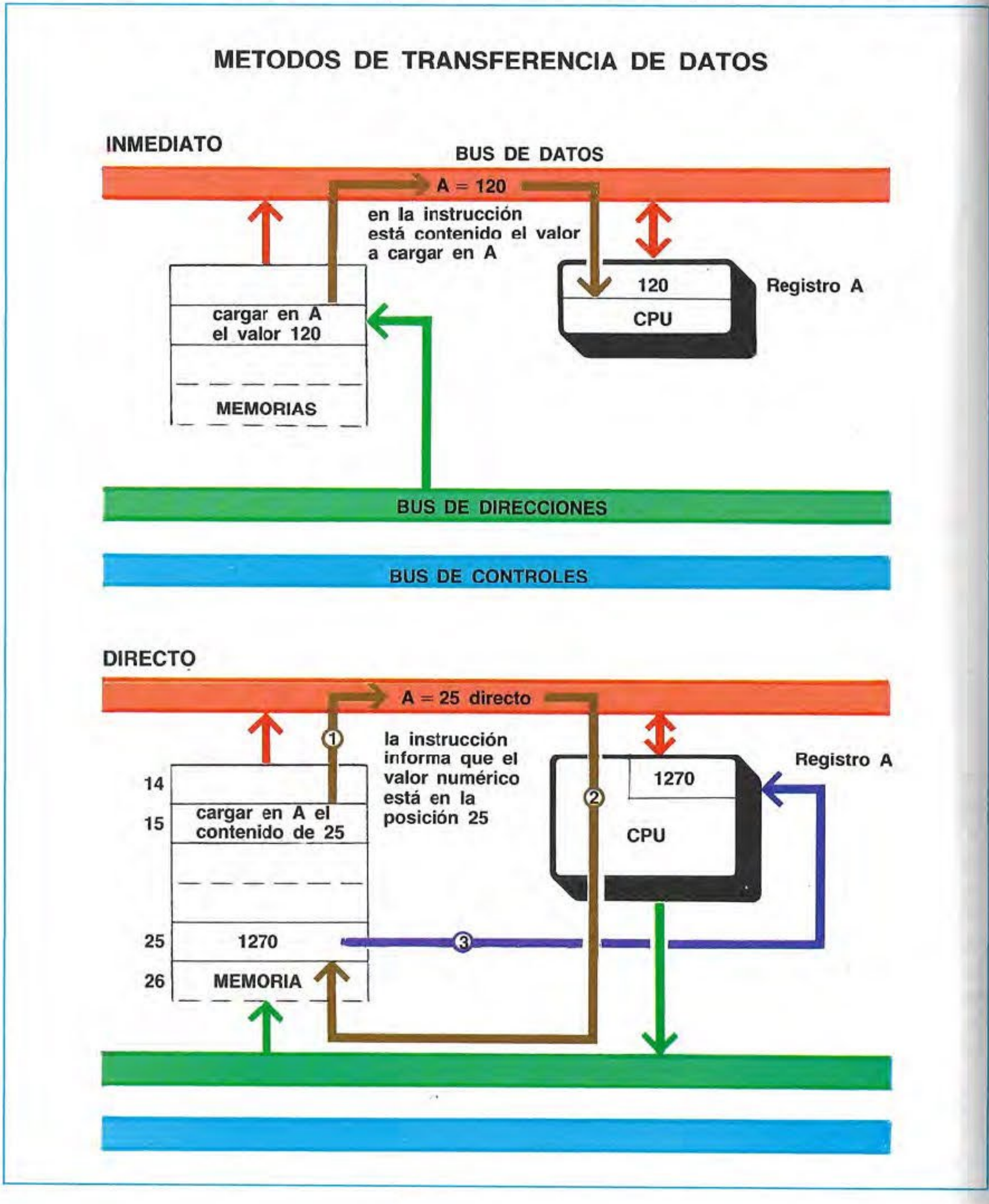
Las instrucciones en la CPU

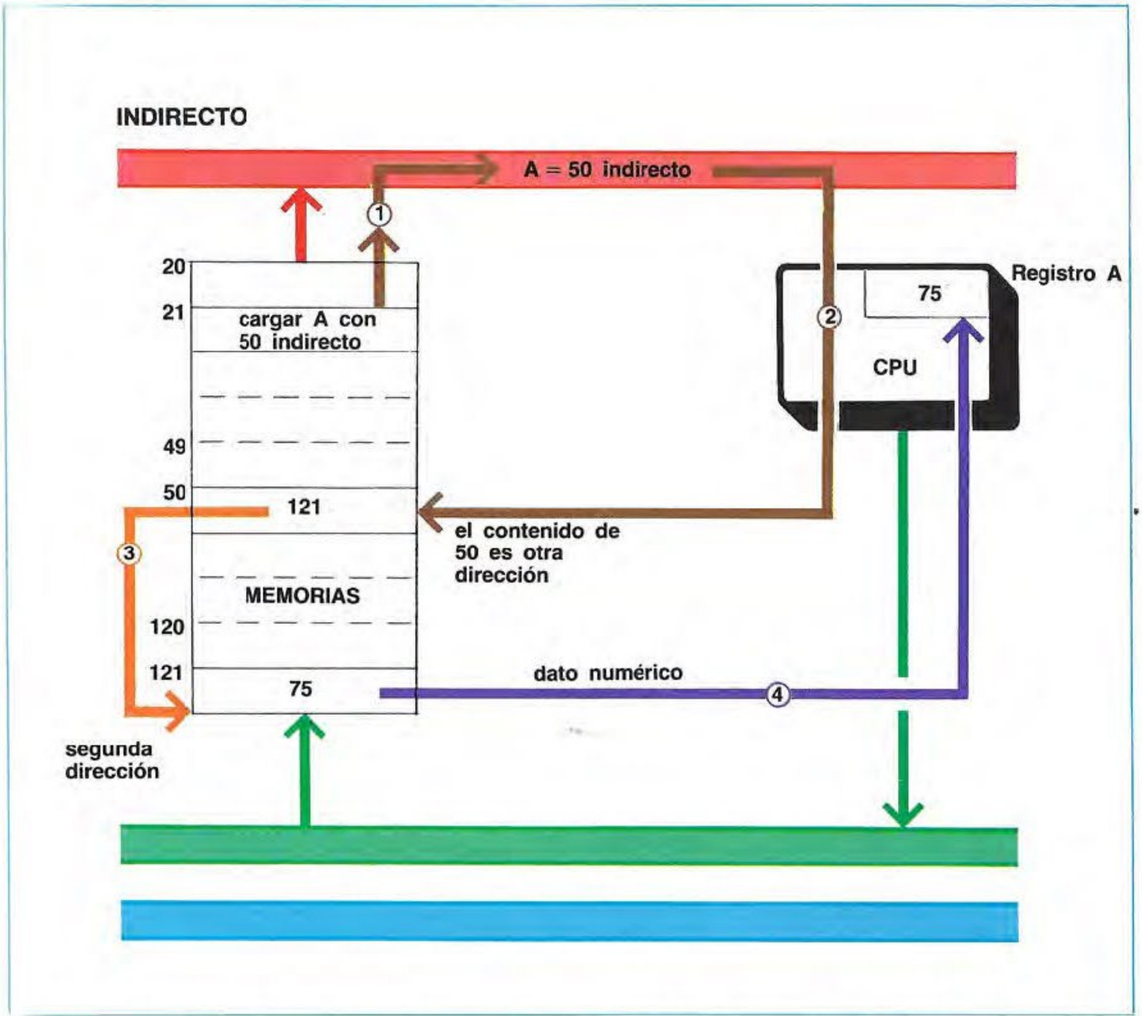
El funcionamiento de un microordenador consiste en la ejecución de la serie de instrucciones que constituyen el programa.

En el interior de la CPU, cada instrucción es interpretada y convertida, por medio de los MICROCÓDIGOS, en señales eléctricas. Las instrucciones que una CPU puede interpretar se dividen en cinco tipos fundamentales:

1 / Instrucciones de transferencia de datos
2 / Instrucciones aritméticas y lógicas
3 / Instrucciones de salto
4 / Instrucciones I/O (Entrada/Salida)
5 / Instrucciones especiales

En función del tipo de CPU, cada uno de los





cinco grupos se compone de un número diferente de instrucciones más o menos complejas.

Instrucciones de transferencia de datos

Son las instrucciones que sirven para sacar o depositar un dato en una memoria.

En los microprocesadores (μP) la transferencia siempre tiene lugar entre una memoria del sistema y una memoria especial, contenida en la CPU, denominada REGISTRO (ver los gráficos adjuntos). Las instrucciones de transferencia de datos pueden ser de tres clases:

Inmediatas: en el registro indicado se carga un valor numérico determinado; por ejemplo: cargar en el registro A el valor 120.

Directas: el registro se carga con el contenido de una memoria determinada; por ejemplo: cargar en el registro A el valor numérico contenido en la memoria 25. Al contrario de lo que ocurre con la instrucción precedente, el registro no se carga con un dato especificado («inmediatamente») sino que lo hace con el contenido de la memoria 25.

Indirectas: constituyen una extensión del tipo anterior. El registro se carga con dos instrucciones sucesivas, de las que la primera suministra la segunda, y la segunda, el dato; por ejemplo: cargar el registro A con 50 indirecto. La ejecución de la instrucción tiene, en este caso, el siguiente desarrollo:

① - ② se toma el contenido de la memoria 50
③ este contenido –por ejemplo, 121– se interpreta como una nueva dirección
④ se toma el dato contenido en la memoria 121

En los ejemplos sólo se han considerado transferencias de datos de la memoria al registro pero, naturalmente, también están previstas las transferencias a la inversa.

Instrucciones aritméticas y lógicas

Este grupo de instrucciones comprende tanto las operaciones aritméticas como las lógicas. Para efectuar una operación hay que suministrar las direcciones de las que tomar los datos y aquella en la que hay que depositar el resultado. La secuencia lógica completa de las instrucciones a suministrar es la siguiente:

① cargar el registro A con el primer dato (contenido, por ejemplo, en la memoria 26)

② efectuar la operación (por ejemplo, la suma) entre el contenido de A (primer dato) y el de la memoria indicada (por ejemplo, 41): el resultado está en A

③ depositar el contenido de A (resultado) en la memoria indicada (por ejemplo, 53)

Instrucciones de salto

Permiten saltar de un punto a otro del programa. Se dividen en dos tipos fundamentales:

- **Instrucciones de salto propiamente dicho (JUMP)**
- **Instrucciones de salto con retorno**

En el primer caso (ver gráfico superior de pág. 149), la ejecución prosigue desde el punto de llegada del salto hasta el final del programa; en el segundo caso, al encontrar la instrucción RETURN, la ejecución vuelve a partir de la instrucción siguiente a la del salto (ver gráfico inferior de pág. 149).

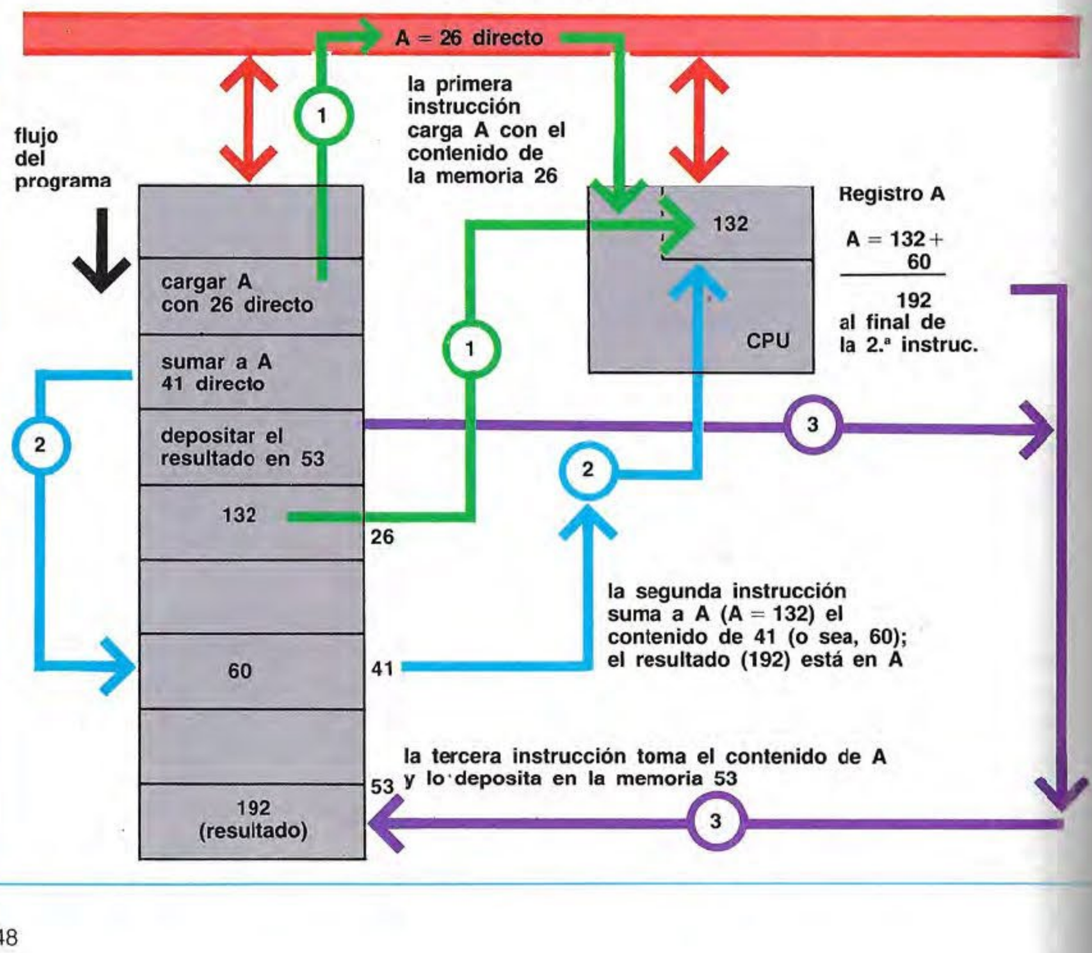
Instrucciones I/O (Entrada/Salida)

Permiten leer o escribir un dato mediante los periféricos.

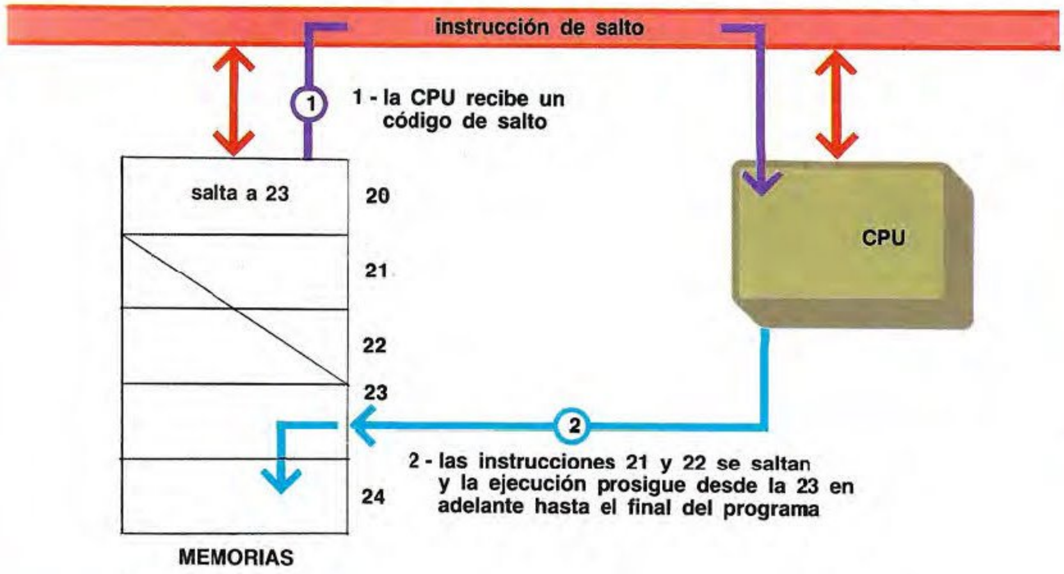
Instrucciones especiales

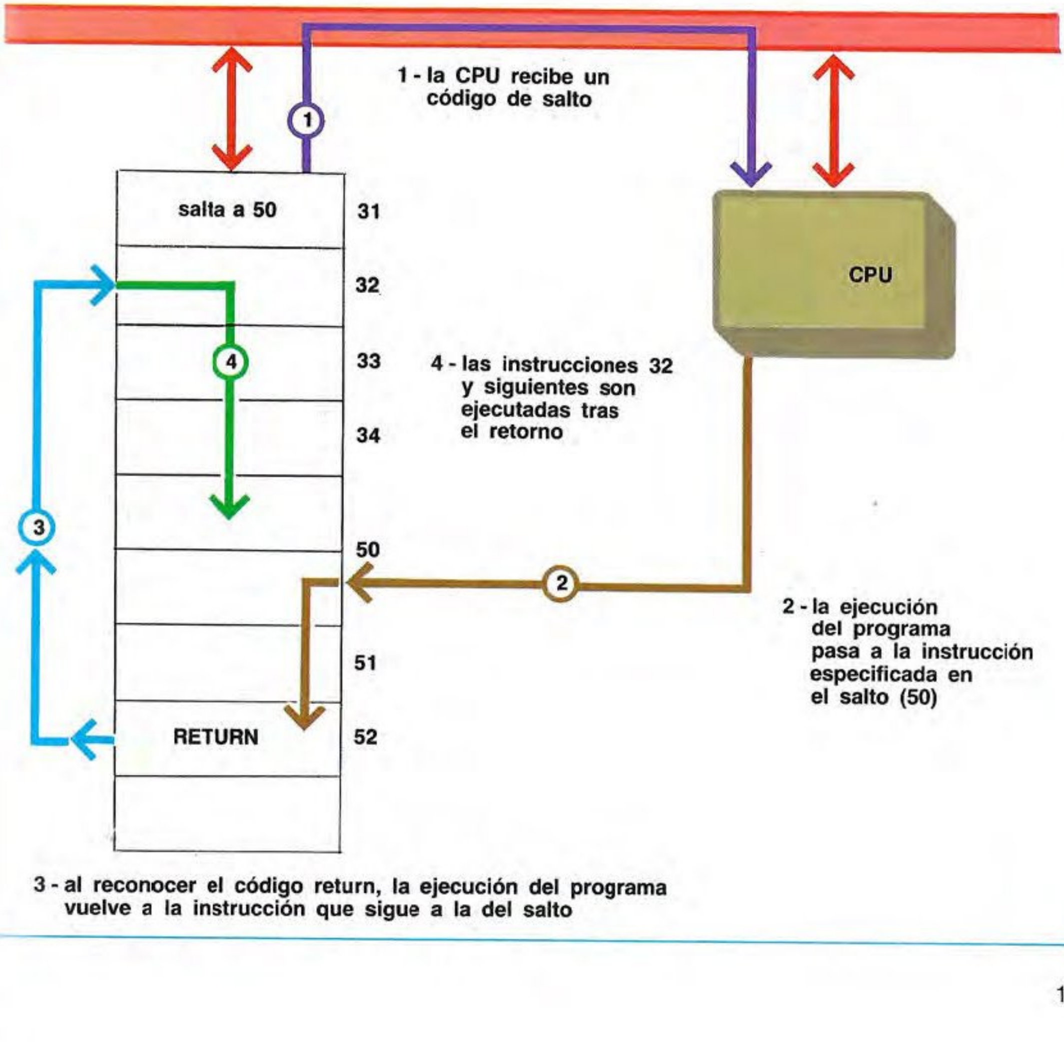
Muchas sólo son inherentes a CPU concretas, y no tienen aplicación general. Las principales tienen que ver con las funciones de Interrupt.

LOGICA DE DESARROLLO DE LAS INSTRUCCIONES ARITMETICAS



LOGICA DE DESARROLLO DE LAS INSTRUCCIONES DE SALTO





SOLUCIONES DEL TEST 3

1 / La traducción en binario como números es:

Decimal	Binario	
7	**0111**	**$(2^2+2^1+2^0=4+2+1)$**
9	**1001**	**$(2^3+2^0=8+1)$**
0	**0000**	**0**
3	**0011**	**$(2^1+2^0=2+1)$**

La traducción en ASCII es:

Decimal	ASCII		
7	37 hexadecimal	=	55 decimal
9	39 hexadecimal	=	57 decimal
0	30 hexadecimal	=	48 decimal
3	33 hexadecimal	=	51 decimal

La primera forma (numérica) se utiliza en los cálculos; la segunda, en las operaciones I/O y de memorización.
El programador ha de especificar, con las adecuadas instrucciones, cuál es la forma que va a utilizar para evitar errores en los cálculos.

2 / La transmisión en serie consiste en enviar los datos (bits) de uno en uno; en la transmisión en paralelo se envía simultáneamente un cierto número de datos (por ej., un carácter completo). La transmisión en paralelo permite mayores velocidades, pero precisa líneas con varios conductores.

3 / El código del carácter K en ASCII es 4B (hexadecimal) =75 (decimal) en binario 1001011. Hay cuatro símbolos 1, luego el bit de paridad es 0.
Para completar el mensaje falta un bit de START (SPACE = 0) y dos bits de STOP (MARK), luego la forma completa es:

0	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
START	←			Letra "K"			→	Pari-dad	STOP	

4 / a) Salto de página = FF = 12 decimal
b) Caracteres X, Y = 88, 89 decimales
c) Salto de línea = LF = 10 decimal
d) x, y minúsculas = 120, 121 decimales
e) Salto de página = FF = 12 decimal

El mensaje completo (en decimal) es:

12	**88**	**89**	**10**	**120**	**121**	**12**
FF	**X**	**Y**	**LF**	**x**	**y**	**FF**
control	carácter		control	carácter		control

Si este mensaje se enviara sin especificar que se trata de caracteres ASCII sería un error, ya que estos datos, entendidos como números en vez de códigos ASCII, carecen de significado para la impresora.

5 / Las letras mayúsculas tienen códigos comprendidos entre 65 y 90 (decimales); las minúsculas, entre 97 y 122. Por tanto, basta con sumar 32 (decimal); efectivamente:

65	(valor decimal de A)
+ 32	TRANSFORMACION
97	(valor decimal de a)

Análogamente:

z (= 122) = Z (90) + 32

6 / Para extraer (con AND) un bit cualquiera hay que utilizar una máscara que contenga el símbolo 1 en la posición a extraer.
La máscara necesaria es:

Bit número	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Máscara	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**

En decimal, esta máscara vale: $2^7 + 2^3 + 2^1 = 128 + 8 + 2 = 138$
Para efectuar la segunda parte, antes hay que pasar los números 12, 21, 6 y 15 a binario (ver tabla 1 de pág. 64).

12 =	0	0	0	0	1	1	0	0
21 =	0	0	0	1	0	1	0	1
6 =	0	0	0	0	0	1	1	0
15 =	0	0	0	0	1	1	1	1

Aplicando el operador AND entre los números y la máscara se obtiene; para el número 12:

Bit número	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Número 12 =	0	0	0	0	1	1	0	0
Máscara =	1	0	0	0	1	0	1	0
12 AND máscara	0	0	0	0	1	0	0	0

Análogamente:

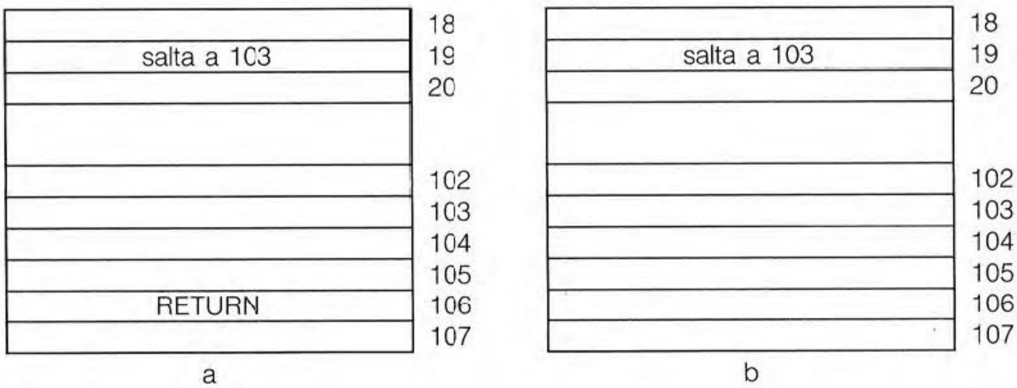
21 AND máscara = 00000000
6 AND máscara = 00000010
15 AND máscara = 00001010

TEST 4

1 / ¿Qué función realiza el Contador de Programa?
- a) Contiene el número de las instrucciones ejecutadas.
- b) Contiene la dirección de la memoria de la que hay que tomar la próxima instrucción a seguir.
- c) Es una memoria auxiliar utilizada para efectuar sumas.

---

2 / En la figura siguiente se representan dos mapas de memoria. ¿Qué instrucción se ejecuta después de la 103 en cada uno de los dos casos?



---

3 / ¿En qué componente de la CPU se efectúan los cálculos aritméticos?

- a) En el STACK.
- b) En la ALU.
- c) En el PC.

---

4 / ¿Qué es un Sistema Operativo?
- a) Un conjunto de programas que realiza funciones básicas tales como la gestión de los periféricos.
- b) Los programas que sirven para efectuar cálculos.
- c) Los componentes especiales que gobiernan el intercambio de datos en DMA.

---

5 / ¿En qué consiste el DMA?
- a) En el acceso directo a la memoria por parte del usuario.
- b) En la modificación de los contenidos de memoria por parte de la CPU.
- c) En un intercambio de datos sin intervención de la CPU.

---

6 / En la figura siguiente se representa un mapa de memoria. El programa comienza en la posición 31; decir si hay un error y, en caso afirmativo, cuál es.

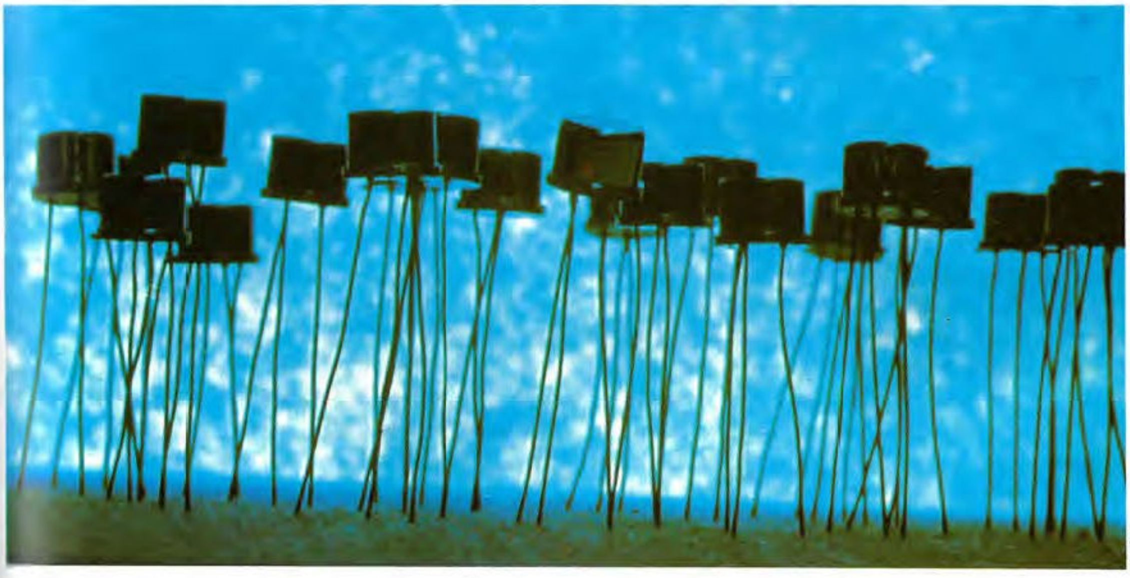
Sumar 12 al contenido de A	31
Salta a 57	32
RETURN	33
	56
RETURN	57
	58
	59

7 / ¿Cuál es la definición correcta de Interrupt?
- a) Interrupción del funcionamiento de la CPU.
- b) Interrupción del programa en función de una demanda externa.
- c) Interrupción de los intercambios con la memoria.

8 / Decir cuáles de estas afirmaciones son correctas:
- a) Una memoria EPROM no puede contener programas.
- b) La unidad de disco contiene alrededor de 64.000 caracteres.
- c) Una memoria ROM no puede ser escrita por el programa.
- d) La CPU se pone en Hold al recibir un Interrupt.
- e) La pantalla y el teclado no pueden generar un Interrupt.
- f) La unidad de disco, normalmente, intercambia datos en DMA.

Las soluciones, en la pág. 183.

Hoy día, en los sistemas digitales, los transistores se sustituyen por circuitos integrados.



Marka

La programación

Escribir un programa consiste en traducir todas las operaciones a efectuar en un lenguaje comprensible para la máquina.
Al plantear una operación en una calculadora corriente, el operador indica, pulsando las teclas, la función a realizar; los circuitos internos interpretan la pulsación de la tecla y generan las señales correspondientes. De manera análoga, pero contraria, las señales que constituyen el resultado son interpretadas por los correspondientes circuitos electrónicos y presentadas, traducidas en números, en el display.
En las máquinas programables, las funciones no se plantean una por una, sino que se «cargan» en la memoria y pueden ser efectuadas en cualquier momento sin más intervención que la orden de arranque. Las tareas que puede realizar una máquina programable son más complejas que el simple cálculo. La función principal del ordenador es la gestión de gran cantidad de datos; la máquina ha de estar en condiciones de asimilar (leer) estos datos, memorizarlos y tratarlos o efectuar búsquedas relativas a los mismos. Todas estas funciones pueden ser activadas por un **Programa**, es decir, por una serie de instrucciones que indican a la máquina qué funciones ha de realizar y en qué orden.
Un programa para la búsqueda del apellido Pérez en una lista, por ejemplo, tendrá que prever las siguientes funciones:

1 / leer un nombre de la lista
2 / si el nombre es Pérez, escribir OK y terminar la tarea
3 / si el nombre no es Pérez, volver al paso 1 (leyendo, pues, el siguiente nombre).

La máquina, sin embargo, no puede interpretar frases descriptivas como estas; por lo tanto, antes de introducir las instrucciones en la memoria hay que traducirlas en una forma que puedan ser interpretadas y convertidas en señales eléctricas.
La forma de expresión más afín a la máquina es la simbología binaria; por el contrario, la forma más afín al hombre es la de suministrar instrucciones que describen el resultado sin detallar la forma de obtenerlo.
Entre estos dos extremos hay numerosos compromisos, es decir, numerosas formas de traducción que permiten obtener una mediación adecuada. Estas formas de expresión se denominan **Lenguajes de programación**.

Los lenguajes de programación

El lenguaje más idóneo para resolver un problema determinado se elige según el tipo de aplicación. En algunos casos dicha elección tiene una importancia secundaria, puesto que se traduce en un mero ahorro de tiempo o del número de instrucciones del programa, por lo que se puede elegir en base a las preferencias personales, aunque sea en detrimento de la eficiencia. En otros casos, sin embargo, el problema no puede resolverse si no se utiliza el lenguaje adecuado.
Todos los lenguajes de programación son «simbólicos», es decir, expresan en una simbología comprensible para el hombre la tarea que la máquina ha de realizar; esta simbología ha de ser traducida a una forma comprensible por la máquina mediante programas especiales: los **Intérpretes** y los **compiladores**.
Además de los lenguajes simbólicos, están los lenguajes «absolutos», en los que las instrucciones se expresan directamente en binario. Los absolutos, por regla general, no se usan para la confección de los programas de aplicación, sino que se utilizan exclusivamente para aplicaciones especiales.
Los lenguajes de programación se dividen en dos tipos fundamentales:

- **lenguajes de alto nivel**
- **lenguajes de bajo nivel**

Cada uno de los dos tipos tiene características y campos de aplicación bien definidos.

Lenguajes de alto nivel

Se denominan «de alto nivel» todos los lenguajes pensados para programador. En estos lenguajes las instrucciones tienen códigos mnemotécnicos que describen, en inglés, la acción a realizar; por ejemplo, para escribir un carácter en la pantalla están las instrucciones PRINT... (lenguaje Basic) o WRITE... (lenguaje Fortran). Ambas instrucciones (PRINT y WRITE) indican la acción a realizar (imprimir, escribir).
El ordenador no puede ejecutar directamente instrucciones en esta forma, puesto que la CPU conoce sólo un cierto número de instrucciones fundamentales (ver «Las instrucciones en la

CPU», pág. 146); por lo tanto, hay que traducir la instrucción del lenguaje de alto nivel a la serie de instrucciones elementales.

Esta traducción puede ser realizada por dos clases distintas de programa, que se denominan Compiladores e Intérpretes.

Los Compiladores son programas de traducción que trabajan sobre todo el programa de usuario y suministran su traducción completa.

Los Intérpretes, normalmente, traducen una sola instrucción a la vez.

Los lenguajes de alto nivel más usados en los micro y miniordenadores son:

■ **Basic** ■ **Fortran** ■ **Cobol**

Otros lenguajes, menos conocidos, se describen en el capítulo dedicado a los Sistemas Operativos. El FORTRAN se utiliza para trabajos científicos; el COBOL, para tareas de gestión y administración; el BASIC es el más versátil, puesto que sirve para ambos tipos de tareas.

El FORTRAN y el COBOL son siempre compilados, mientras que del BASIC existen las dos formas: interpretado y compilado.

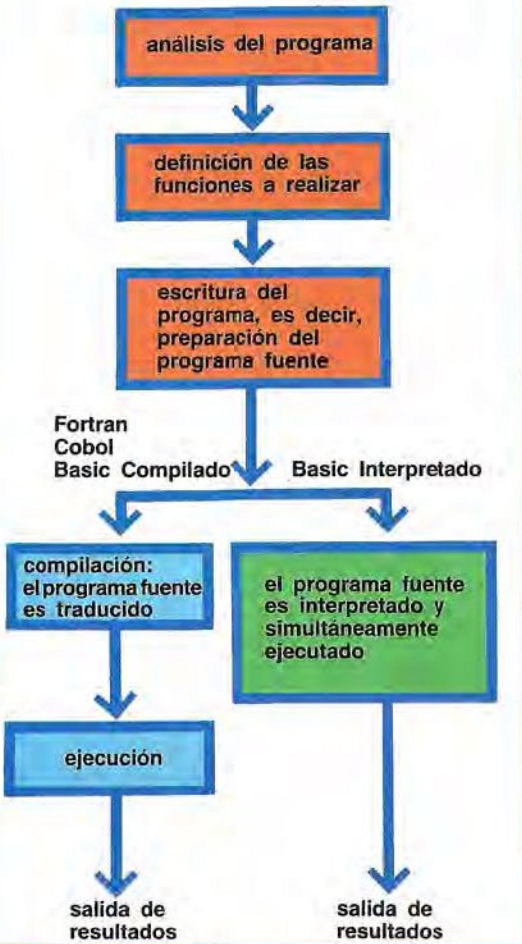
Un programa de usuario escrito en un lenguaje cualquiera se denomina **Fuente**.

En el gráfico contiguo se representan de forma esquemática las distintas fases en que se articula el procedimiento de preparación y ejecución de un programa en el caso de que se adopte un lenguaje compilado o un lenguaje interpretado.

En los lenguajes interpretados (Basic Interpretado), cada instrucción del programa fuente es traducida (interpretada) en el momento de su ejecución. Puesto que mientras se desarrolla el programa ha de efectuarse también la traducción de las instrucciones, la ejecución resulta más larga. Los lenguajes interpretados tienen, sin embargo, la ventaja del diagnóstico en el momento de la introducción del programa fuente. El programa fuente se introduce pulsando las instrucciones en el teclado, y el Sistema Operativo lo carga en la memoria. Durante esta fase, si hay errores de algún tipo, el Intérprete los señala y permite su corrección inmediata. Por el contrario, en los lenguajes compilados los errores se evidencian sólo en el momento de la ejecución del programa. Con los lenguajes interpretados, los errores banales de tecleado son evidenciados y corregidos inmediatamente; sin embargo, con los lenguajes compilados hay que esperar todo el tiempo de la compilación.

Por tanto, en los lenguajes interpretados la co-

ESQUEMA LOGICO DEL PROCESO DE ESCRITURA Y EJECUCION DE UN PROGRAMA



rrección de un error es más rápida, puesto que no hay que esperar el tiempo necesario para las operaciones de compilación, es decir, no hay que esperar todo el tiempo de traducción del programa entero.

Resumiendo, las principales características de los dos tipos de lenguaje son:

Lenguaje Interpretado: posibilidad de corrección inmediata; ejecución más lenta.

Lenguaje Compilado: cada corrección implica una nueva compilación y, por tanto, un tiempo de espera; ejecución más rápida.

Lo ideal es disponer de un lenguaje que pueda ser tanto interpretado como compilado.

Al escribir un programa es muy útil tener un Intérprete, puesto que así se pueden efectuar correcciones y pruebas sin perder el tiempo de la

compilación. Al final de la fase de prueba, cuando el programa está correcto, se compila para tener el programa en su forma más rápida.
Por su versatilidad, el Basic se presta a esta metodología, y por eso es el lenguaje más utilizado en los micro y miniordenadores. Además, por haber sido concebido para principiantes, es el más apto para iniciarse en la programación.

Compilación

El procedimiento de compilación de un programa fuente consta de dos pasos sucesivos:

- el primer paso consiste en la producción de una serie de módulos intermedios denominados **Reubicables**.
- el segundo paso consiste en «enlazar» entre sí estos módulos; esta operación se llama **Link**, tras la cual se obtiene el denominado **Programa objeto**, que puede ser cargado en la memoria y ejecutado.

ESQUEMA DE LAS FASES DE COMPILACION



Normalmente, en los sistemas dotados de compilador hay que disponer de unidades de disco para memorizar los módulos intermedios; por el contrario, en los lenguajes interpretados aquéllas no son indispensables, ya que el intérprete puede residir permanentemente en la memoria.

Basic

Es un lenguaje apto para objetivos científicos y comerciales, ya que, además de las características del FORTRAN, tiene una notable capacidad para gestionar textos y diversas instrucciones para la expresión de informes económicos. La gestión de datos en disco no es tan potente como la del COBOL, pero es adecuada a la estructura de los micro y miniordenadores.
El BASIC, con respecto a los otros lenguajes, es bastante reciente y aún no ha sido adoptado por todos de la misma forma. Según el fabricante, puede tener más o menos instrucciones o una simbología distinta.

Fortran

Es un lenguaje concebido para uso científico. Sus principales características son:
— necesidad de pocas instrucciones fundamentales;
— escasa necesidad de tratar con textos;
— los datos son normalmente números en forma exponencial;
— la cantidad de datos a elaborar es reducida, mientras que puede ser notable el número de elaboraciones necesarias;
— necesidad de disponer de instrumentos de cálculo especiales (algoritmos matemáticos).
La utilización de este lenguaje para fines comerciales o de gestión no debe excluirse a priori, aunque resulte incómodo y trabajoso, sobre todo por la falta de instrucciones que permitan elaborar caracteres alfanuméricos.

Cobol

Es un lenguaje concebido exclusivamente para fines comerciales. Sus características son:
— ausencia de instrucciones de cálculo (excepto las operaciones principales);
— posibilidad de una buena gestión de los datos en disco (archivos);
— instrucciones destinadas a la impresión de informes de tipo económico.

Análisis y diagramas de flujo

Escribir las instrucciones que constituyen un programa es sólo la última fase, la conclusión de un complejo trabajo de análisis del problema específico y de su síntesis en una estructura compatible con la máquina. El análisis del problema a resolver lleva a sintetizar incluso las operaciones más complejas en una serie de funciones elementales representables gráficamente mediante los símbolos adecuados. De esta representación gráfica (flow-chart: diagrama o carta de flujo) se pasa a la escritura de las instrucciones propiamente dicha. En este capítulo se ilustran con múltiples ejemplos las principales metodologías de análisis y las técnicas de preparación de los diagramas de flujo.

Planteamiento de un programa

Para cada problema de aplicación concreto hay que suministrar al ordenador el programa adecuado. En las aplicaciones más complejas, para obtener el resultado final hacen falta diversos programas, cuyo conjunto se denomina **procedimiento**.

Por ejemplo, en la gestión de un almacén se pueden distinguir tres aspectos fundamentales:

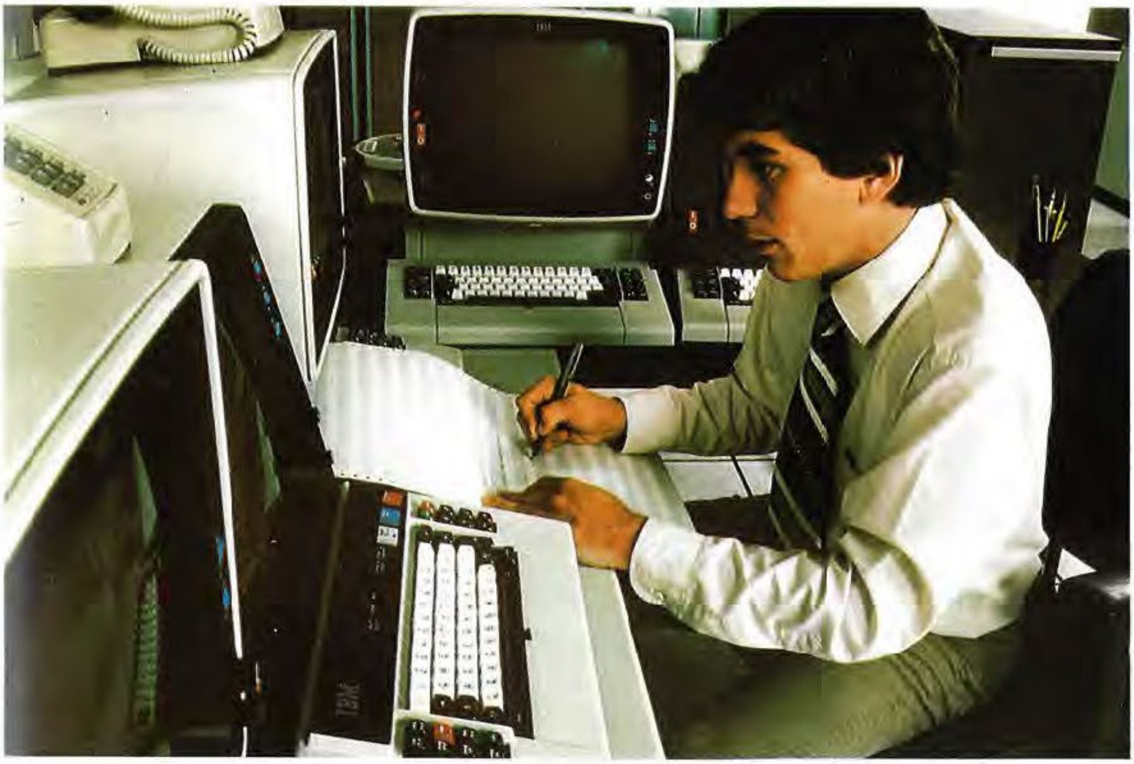
- **entrada de mercancía**
- **salida de mercancía**
- **inventario de existencias**

Cada una de estas funciones es desarrollada por un programa, y su conjunto es el procedimiento de gestión del almacén.

Antes de iniciar la escritura de un programa hay que conocer los aspectos del problema y el método a seguir para resolverlo. Esta fase **(planteamiento)** es la más delicada, puesto que un error de evaluación puede dar lugar a resultados negativos o incompletos. El planteamiento de un programa se puede dividir en tres fases:

- **análisis del problema**
- **síntesis**
- **diagramación (cartas de flujo)**

En las compañías de seguros, la gran capacidad de memorización y la velocidad de búsqueda del ordenador permiten una rápida ejecución de los trámites.



Marka

Análisis

Un programa cualquiera sólo puede elaborar datos que hayan sido suministrados a la máquina. En el ejemplo anterior (inventario de existencias), el programa sólo puede dar resultados correctos si han sido introducidos todos los movimientos de entrada y salida de mercancía; además, cada uno de éstos ha de referirse a artículos que formen parte del almacén, es decir, a los artículos conocidos por la máquina.
El primer paso del análisis consiste en examinar los resultados que se desea obtener y comprobar que se dispone de todos los datos necesarios para ello. Por ejemplo, supongamos que el inventario de existencias deba consistir en una lista con los siguientes datos:

1 / Código del artículo
2 / Descripción
3 / Existencias
4 / Costo unitario
5 / Valor total de las existencias

Analizando las salidas requeridas se establecen los diversos programas que formarán parte del procedimiento, cuyo esquema lógico se muestra en el gráfico contiguo.
En el ejemplo propuesto, los primeros datos a imprimir son:

1 / Código del artículo
2 / Descripción

Para disponer de estos datos hay que haberlos introducido, por lo que ha de haber un programa para la creación de las denominaciones de almacén.
La demanda siguiente (3 / Existencias) es la impresión, repetida para cada artículo, de la cantidad existente en el almacén.
Este dato puede variar por dos razones:

- **salida de un artículo (descarga)**
- **llegada de un artículo (carga)**

Por lo tanto, hay que prever un programa que para caca salida reste de las existencias lo que sale, y un programa que, para cada entrada, sume a las existencias lo que ha llegado.
La denominación 4 (Costo unitario) es un dato que ha de ser introducido en el ordenador en el momento de la llegada de la mercancía (es decir, en el programa que actualiza las existencias en la fase de carga).

ESQUEMA LOGICO DEL PROCEDIMIENTO DE GESTION DE UN ALMACEN



Por lo tanto, para satisfacer los requisitos 3 y 4 hacen falta los programas:

- **descarga de mercancía (salidas)**
- **carga de mercancía (entradas)**

Por último, la denominación 5 requiere un programa que, utilizando los catos antes introducidos, efectúe los cálculos. El resultado del análisis se puede resumir en una tabla en la que se listan las principales funciones a realizar:

1 / Creación de las denominaciones de almacén (código del artículo y descripción)
2 / Movimientos de entrada (cálculo de las existencias actuales)
3 / Movimientos de salida (cálculo de las existencias actuales)
4 / Impresión de listados de existencias (cálculo del valor total)

Los programas necesarios y los datos con que trabajan se enumeran en la tabla siguiente:

PROGRAMA	DATOS
Creación de denominaciones de almacén	Código del artículo y descripción
Carga	Existencias y precio unitario
Descarga	Existencias
Impresión de listados	Valor total

Funciones principales de cada programa son:

Creación de las denominaciones de almacén
Ha de permitir al usuario introducir un código para cada artículo del almacén, es decir, una sigla que identifique el artículo y su descripción.

Carga
Su objetivo principal es el de sumar cada nueva entrada y actualizar el costo. Las operaciones que ha de realizar son:
Cantidad actual = Cantidad anterior + Cantidad de entrada
Costo = Costo del artículo de entrada

Descarga
Sólo ha de actualizar las existencias al producirse una salida:
Cantidad actual = Cantidad anterior – Cantidad de salida

Impresión de listados
Este programa, antes de proceder a la impresión del contenido del almacén, ha de calcular, para cada artículo, el valor total:
Valor total = Existencias × Costo unitario, es decir: Denominación 5 = Denominación 3 × Denominación 4

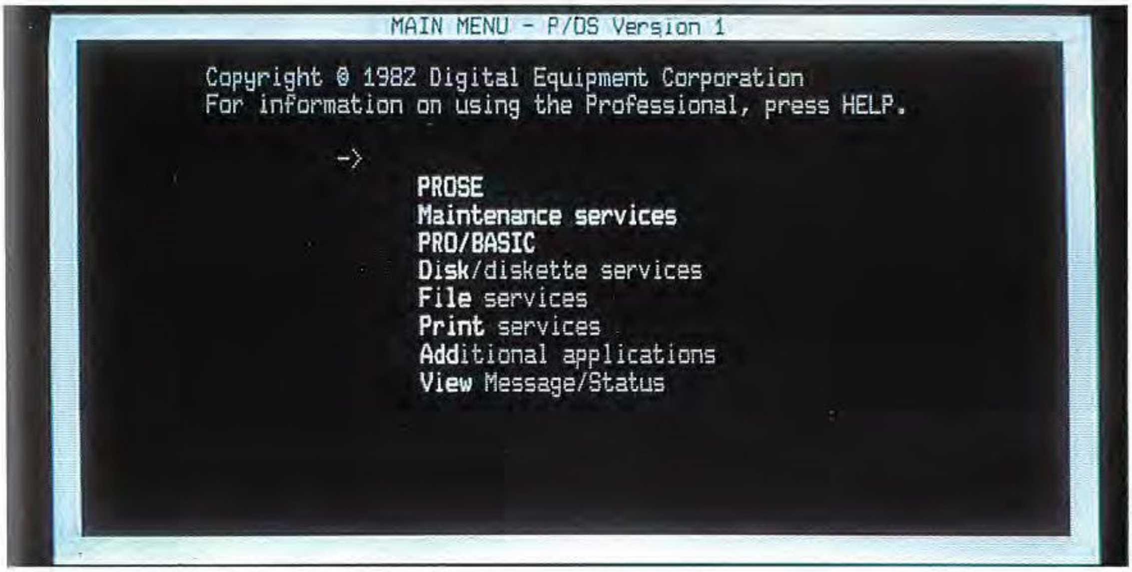
Estas operaciones se esquematizan en los gráficos de las págs. 160 y 161. Los datos relativos a cada denominación han de ser memorizados, para lo cual se necesita una memoria de masa (diskette). La capacidad de memorización, la velocidad y el número de usuarios que han de trabajar simultáneamente definen el tipo de máquina que habrá que usar. La elección definitiva se efectuará solo tras haber completado la fase preliminar (Análisis, Síntesis, Diagramación), pero, en principio, cabe elegir entre:

Ordenador personal. Un solo usuario. Datos a memorizar en número limitado. Ninguna necesidad especial de velocidad. Costo mínimo. Escasas o nulas posibilidades de expansión.

Microordenador. Un solo usuario o, excepcionalmente, varios en número limitado. Datos de cierta consistencia. Posibilidades de lograr una buena velocidad de elaboración. Costo intermedio. Moderadas posibilidades de expansión.

Miniordenador. Posibilidad de varios usuarios simultáneos, incluso con programas distintos. Cantidad de datos importante. Velocidades medias-altas. Costo más bien elevado. Notables posibilidades de expansión.

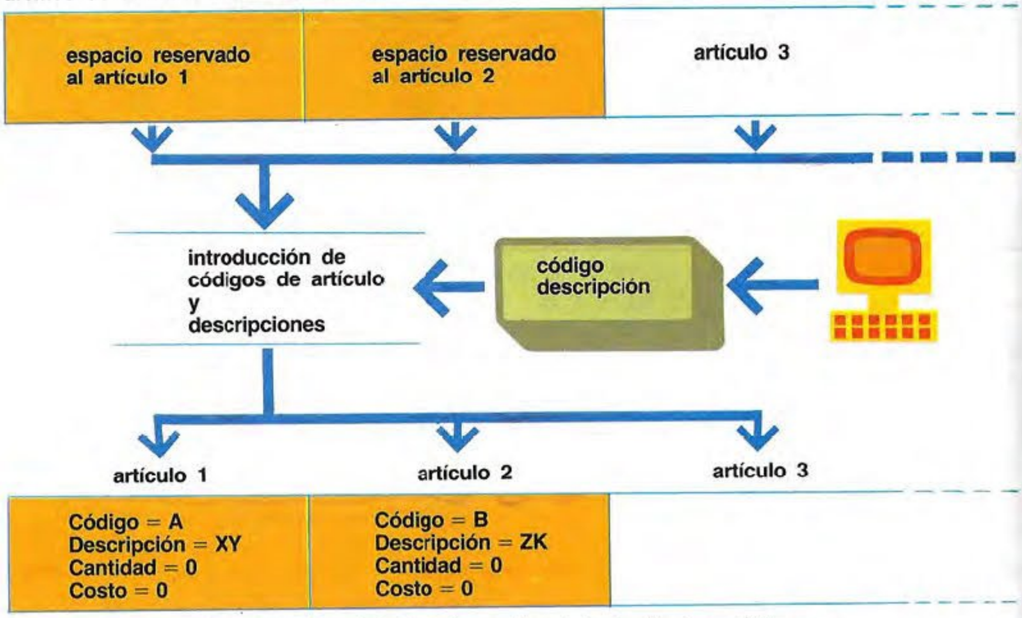
Informaciones visualizadas en la pantalla de un ordenador.



Centro THC

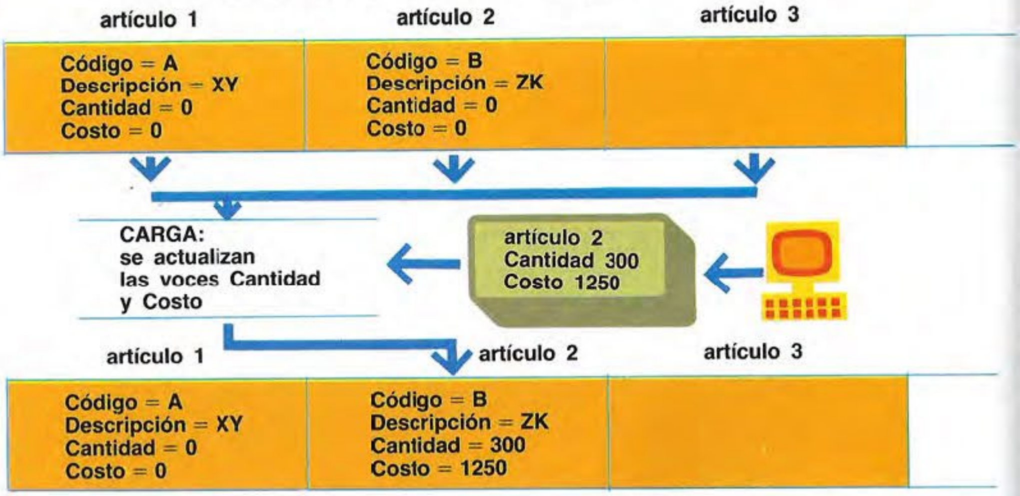
CREACION DE DENOMINACIONES DE ALMACEN

archivo de almacén inicialmente vacío

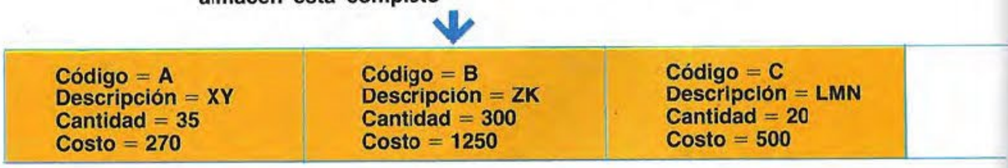


En el archivo del almacén se han introducido los códigos y las descripciones de cada artículo; la cantidad y los costos son iguales a cero, puesto que aún no se ha efectuado la carga

OPERACIONES DE CARGA DE MERCANCIA



La operación de carga (por ejemplo, del artículo 2) actualiza la cantidad (300) y el costo (1250); a la llegada de todos los artículos el archivo de almacén está completo



OPERACIONES DE DESCARGA DE MERCANCIA

Código = A Descripción = XY Cantidad = 35 Costo = 270	Código = B Descripción = ZK Cantidad = 300 Costo = 1250	Código = C Descripción = LMN Cantidad = 20 Costo = 500

DESCARGA:
se actualiza
la denominación
Cantidad

artículo 3
Código C
Cantidad
que sale = 6



Existencias = Cant. anterior-Cant. salida
Existencias = 20 − 6 = 14

Código = A	Código = B	Código = C Descripción = LMN Cantidad = 14 Costo = 500

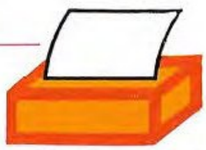
La operación de descarga (salida) resta
de las existencias la cantidad que sale
y actualiza los datos contenidos en el archivo

IMPRESION DE EXISTENCIAS

Código = A Descripción = XY Cantidad = 35 Costo = 270	Código = B Descripción = ZK Cantidad = 300 Costo = 1250	Código = C Descripción = LMN Cantidad = 14 Costo = 500

Se lee el archivo entero
Para cada artículo se
calcula el valor total:
Valor total = Costo × Cantidad

Los valores totales de cada
artículo se suman
y suministran el total
general



Código	Descripción	Cantidad	Costo	Valor total
A	XY	35	270	9450
B	ZK	300	1250	375000
C	LMN	14	500	7000
			Total General	391450

La elección también se habrá de efectuar en base a posibles necesidades futuras.

Los objetivos de la fase de análisis son:
1 / Control de la existencia de todos los datos de entrada
2 / Definición de eventuales programas accesorios
3 / Estimación del tipo de máquina

Síntesis

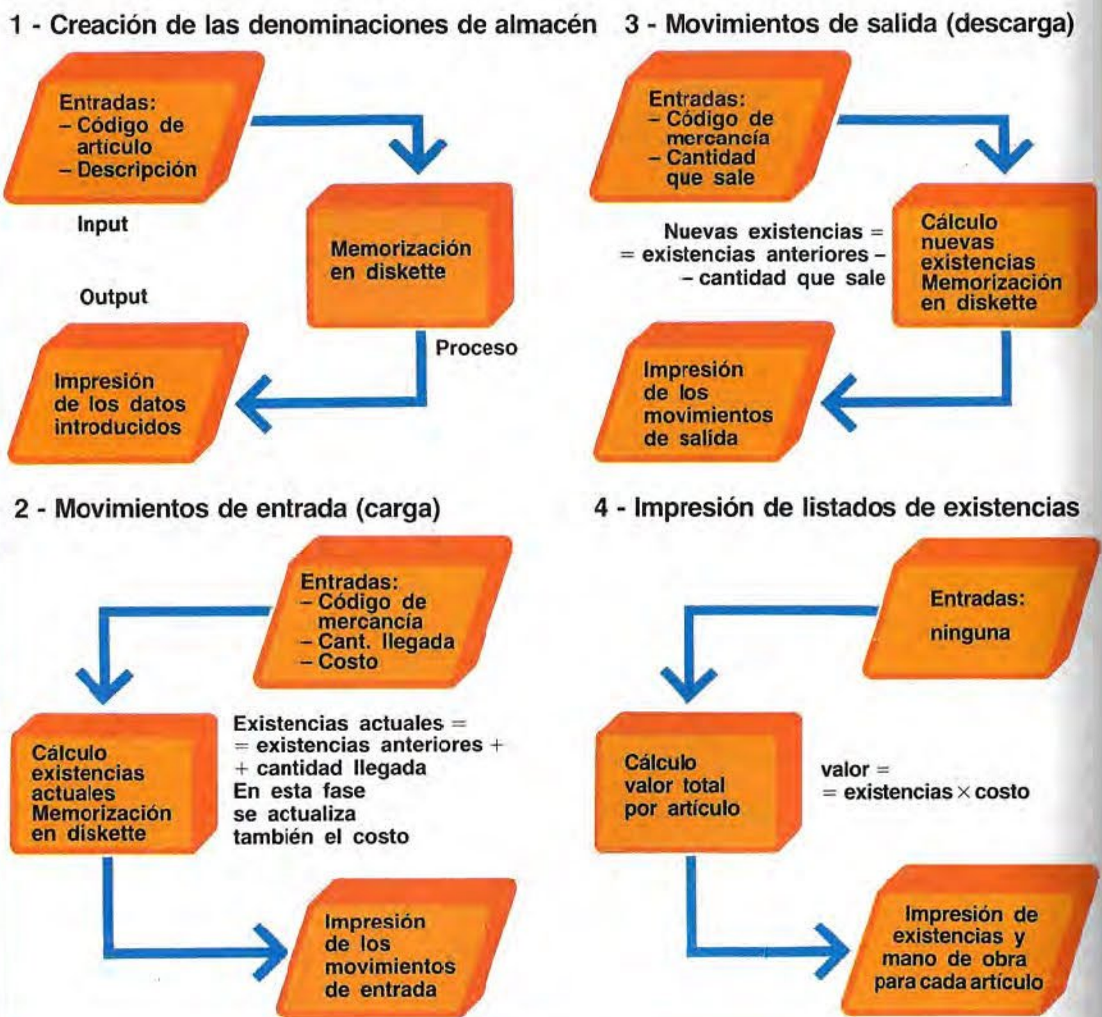
La síntesis consiste en preparar uno o varios esquemas que muestren de forma sinóptica las funciones de un programa o de los programas de un procedimiento. Lo más indicado es utilizar la técnica HIPO (Hierarchy plus Input-Process-Output).

Esta técnica consiste en esquematizar cada programa (o una parte) en tres bloques:

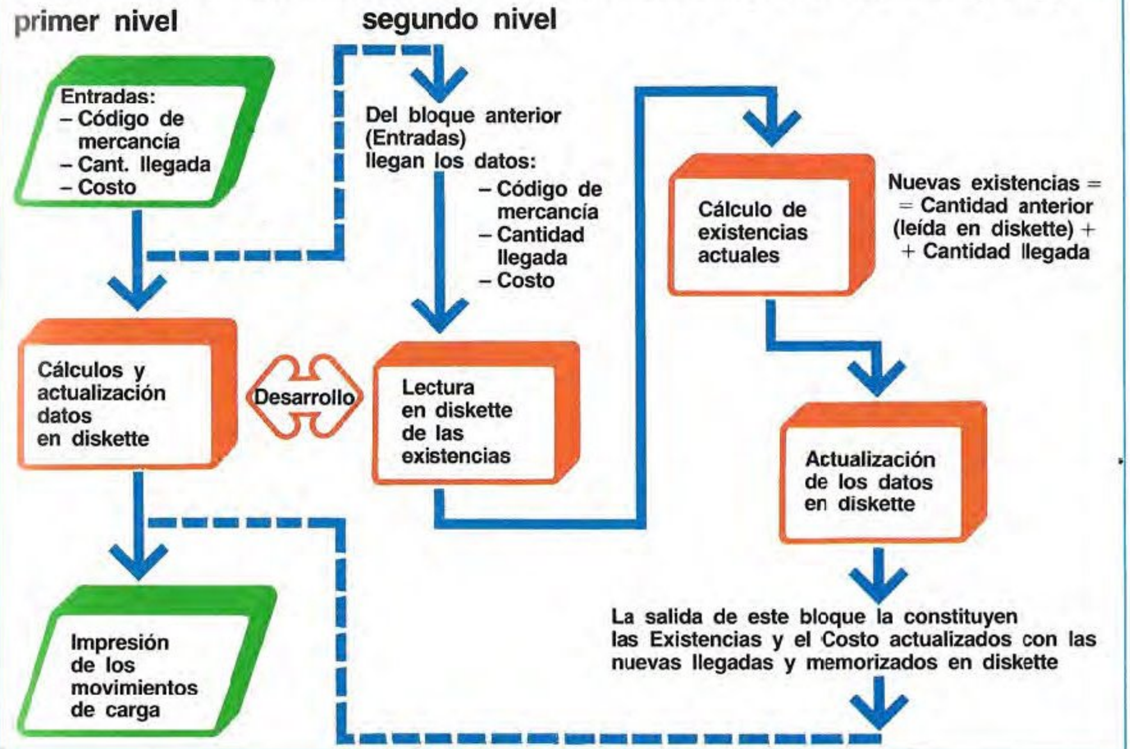
Primer bloque: Input (entrada). Contiene la descripción de todas las entradas necesarias para el programa.

Segundo bloque: Proceso de elaboración. Contiene la descripción de todos los cálculos o,

DOCUMENTACION DEL PROCEDIMIENTO DE GESTION DE UN ALMACEN



EJEMPLO DE DESARROLLO DE UN ESQUEMA DE CARGA DE MERCANCIA



en general, de todos los procedimientos con los que se elaboran los datos.

Tercer bloque: Output (salida). Contiene la descripción de todas las salidas previstas.

En los programas más sencillos puede ser suficiente un solo diagrama HIPO; en los más complejos hacen falta varios diagramas con diversos grados de detalle. La técnica HIPO, además de ser una importante ayuda en el desarrollo del trabajo, constituye la mejor forma de documentación de los programas. En los gráficos de la pág. 162 se muestran los diagramas HIPO del procedimiento de gestión de un almacén.

Otro paso consiste en desarrollar los bloques más complejos. Por ejemplo, en la parte «movimientos de entrada», el bloque Process del programa se puede desarrollar así:

a / lectura en disco, para cada código de mercancía, de las existencias
b / adición, a las antiguas existencias, de la cantidad que llega
c / sustitución del antiguo costo unitario por el nuevo valor
d / sustitución, en disco, de los datos antiguos por los nuevos

Cada una de estas elaboraciones se refiere a un artículo determinado, luego el dato de entrada es el código del artículo.

Al final de la elaboración se tienen en disco los datos actualizados, luego esta es la salida de la elaboración.

En el gráfico superior se muestra el esquema del primer nivel y el desarrollo (segundo nivel). Con el segundo nivel de documentación todavía no se tiene el detalle suficiente para proceder a la escritura del programa; hace falta un nivel con mayor definición.

Este nuevo nivel consiste en preparar un gráfico que ilustre con detalle, utilizando determinados símbolos preestablecidos, todas las operaciones a efectuar.

En el gráfico superior conviene prever en cada entrada un control de validez del dato (que indique, por ejemplo, si se intenta introducir un código de mercancía que no existe); en las salidas a impresora hay que prever algunas líneas de encabezamiento, etc.

El gráfico de las operaciones a realizar, con este nivel de detalle, se denomina FLOW-CHART (Diagrama de Flujo) y es el último paso previo a la escritura del programa.

Las máquinas que leen

Considerando el hecho de que los ordenadores influyen ya en tantos aspectos de nuestra vida cotidiana, resulta sorprendente que haya transcurrido tanto tiempo antes de que aprendieran a leer.

Las aplicaciones que, potencialmente, requieren la adopción de dispositivos para la lectura automática son innumerables.

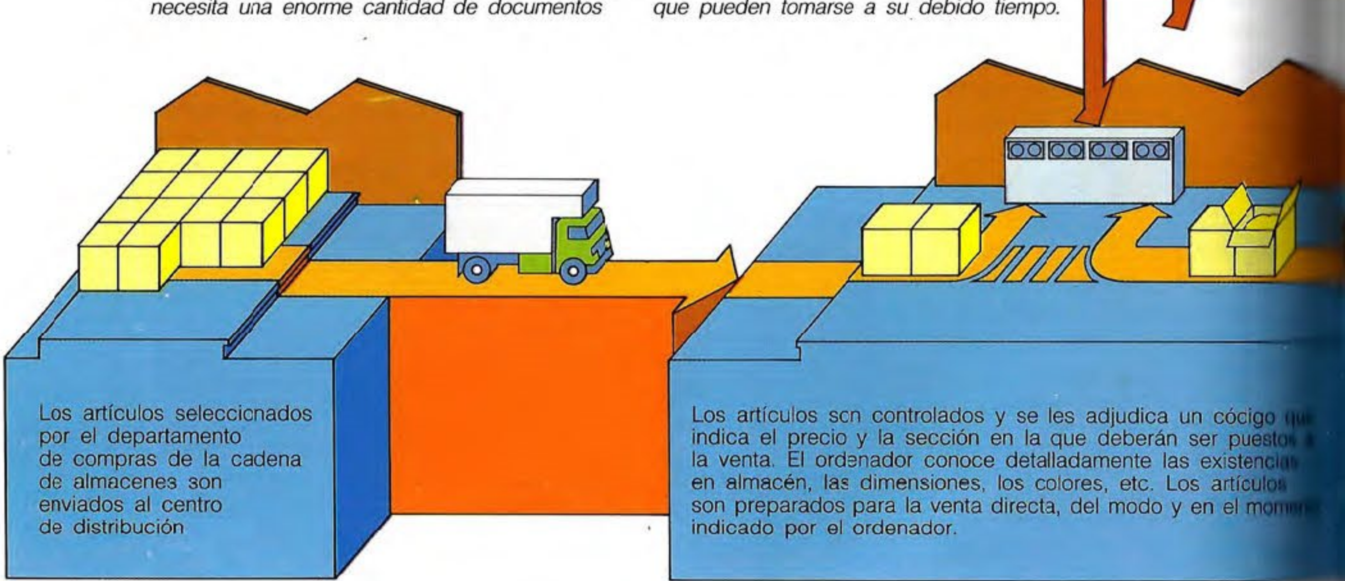
Cada día se escriben cientos de miles de talones bancarios, que han de ser clasificados de forma que cada talón regrese a la agencia donde está depositada la cuenta de la que hay que descontar el importe. Máquinas especiales leen los caracteres impresos con tinta magnética en el borde inferior de los talones y utilizan la información para accionar un veloz sistema de selección automática.

En algunos países el volumen del correo es aún lo suficientemente moderado como para permitir que las direcciones y códigos postales sean leídos por seres humanos. En una oficina de Correos del viejo tipo, un empleado lee la dirección, se acerca a un casillero y coloca el sobre en la casilla correspondiente al número del código postal. En una oficina de Correos automatizada, el empleado lee la dirección y luego marca el código postal en un teclado similar al de una máquina de escribir, y así acciona una máquina que coloca automáticamente el sobre en la casilla correspondiente.

En Japón y Estados Unidos, el volumen del correo es tal y la disponibilidad de mano de obra tan limitada, que para gobernar las seleccionadoras mecánicas se utilizan máquinas lectoras en vez de personas, a pesar de que una de estas máquinas rechaza normalmente el 8 por ciento de los sobres porque no consigue leer el código.

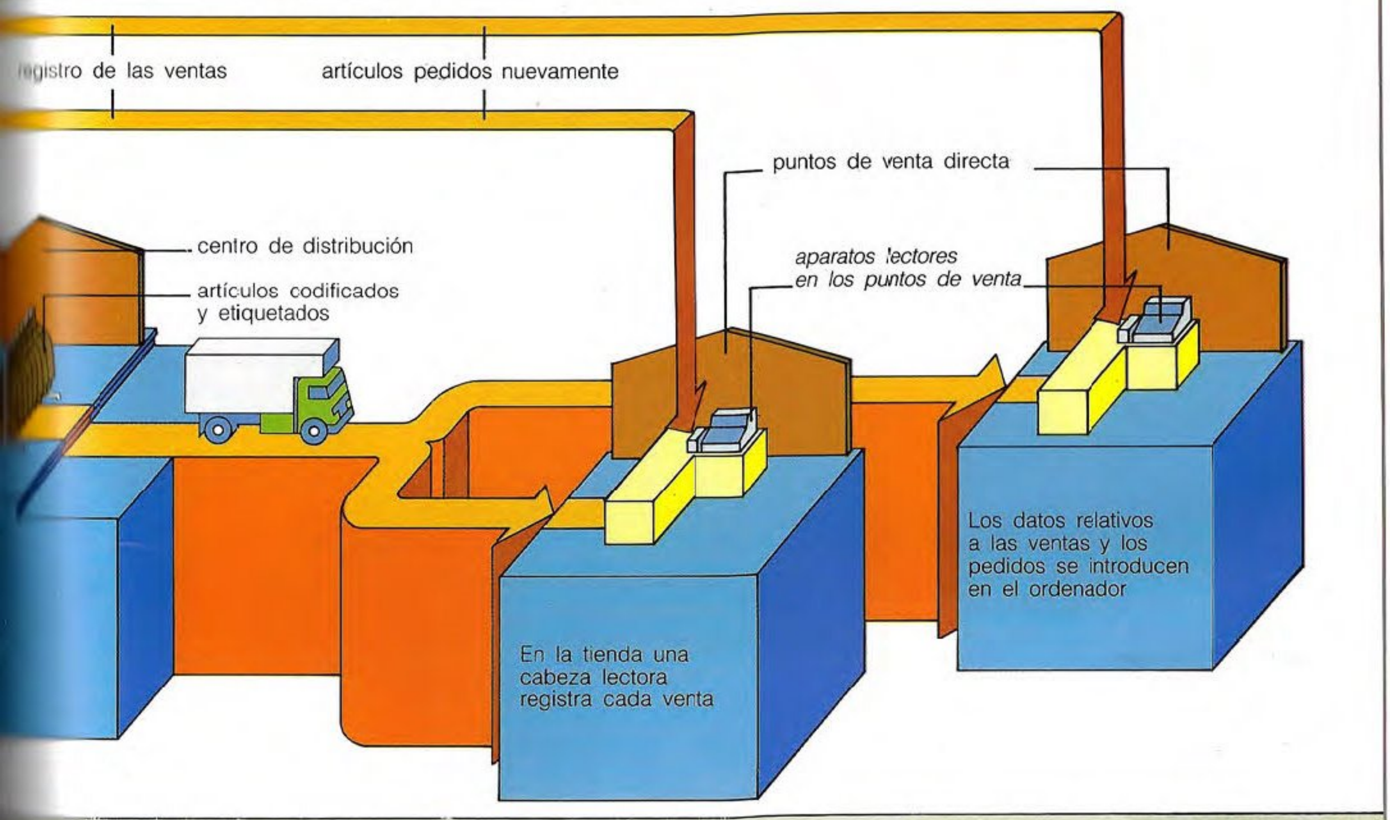
Además de clasificar documentos, las máquinas lectoras pueden introducir datos en los ordenadores. Por ejemplo, para mantener actualizados los nombres y direcciones de un millón de personas y entidades, hay que registrar alrededor de 25.000 cambios a la semana. Las direcciones pueden suministrarse al ordenador mediante el teclado de una consola, que le transmite el texto directamente. Pero también se pueden escribir las direcciones en una hoja de papel, con una simple máquina de escribir, para luego pasar la hoja a la máquina lectora.

En este ejemplo típico de comercialización se puede usar un gran número de funciones de lectura óptica de caracteres y contraseñas. La ventaja estriba en la reducción del tiempo necesario para preparar los datos a insertar en el ordenador central. En un gran almacén se necesita una enorme cantidad de documentos diversos para ordenar, tratar, controlar, y vender los distintos artículos. Los datos pueden estar escritos a mano, a máquina, impresos o codificados. Las máquinas lectoras traducen los datos al lenguaje del ordenador y los almacenan en una cinta magnética, de la que pueden tomarse a su debido tiempo.



Las oficinas de correos de Gran Bretaña utilizan máquinas capaces de leer caracteres, alfabéticos, numéricos y de puntuación al ritmo de unos 2.400 caracteres por segundo (los tipos son seleccionados cuidadosamente por su forma y tamaño). Una máquina fabricada para el Bank of América puede leer 40.000 caracteres por segundo, pero el mero hecho de lograr una velocidad de unos 1.000 caracteres por segundo requiere equipos complicados y costosos.
Las máquinas lectoras se utilizan también para introducir textos en los procesadores de palabras. Un mecanógrafo compone el texto con una máquina especial, el autor lo repasa y efectúa las correcciones pertinentes, y luego una máquina lectora transmite el texto al procesador de palabras, que permite al operador aportar los ajustes necesarios con gran rapidez, evitando que el texto lleve tachaduras. Desgraciadamente, las máquinas lectoras poco costosas utilizadas con los procesadores de textos, no son fiables cuando se trata de leer un texto que no haya sido escrito con caracteres especiales, por ejemplo, correspondencia con correcciones. Los textos de escritura especial requieren intervalos claros entre los caracteres.

Las compañías de gas y electricidad utilizan máquinas lectoras para descifrar una serie de números que hay en cada factura.
Estos números dicen al ordenador con toda exactitud qué factura ha sido pagada.
Una máquina de alta velocidad que pueda leer letras del alfabeto, números y signos de puntuación de distintos tipos, exige una gran inversión. Para reducir costos hay que hacer que los datos sean leídos por la máquina con facilidad y exactitud. Se puede lograr una notable reducción de costos haciendo que la máquina lea, no caracteres, sino secuencias codificadas de finas líneas paralelas, líneas mas gruesas y espacios. Estos códigos (códigos de barras), que pueden ser leídos por un aparato portátil, se han hecho de uso común en bibliotecas y almacenes, ya que los aparatos portátiles para la lectura de los datos codificados que se usan en librerías y supermercados son mucho más sencillos y baratos que los sistemas portátiles capaces de leer los caracteres normales.
En relación al hombre, las máquinas están concebidas para poder tolerar menos variaciones en la forma y en la calidad de los caracteres, pero, en cambio, son capaces de identificar un



número considerablemente mayor de caracteres por segundo.
El costo de una máquina lectora depende de la gama de caracteres que ha de leer y de la velocidad de lectura. Los progresos de la microelectrónica han hecho que la parte procesadora de la máquina sea relativamente barata con respecto a las partes mecánicas que mueven físicamente el papel. Para reducir costos, algunas máquinas recientes son poco sofisticadas desde el punto de vista mecánico y sólo pueden leer unos diez caracteres por segundo. Una solución especialmente económica consiste en pasar manualmente una cabeza lectora sobre la línea del texto que ha de ser leído.
Si la tinta, el papel y el tipo de impresión responden exactamente a las especificaciones del fabricante, una buena máquina puede leer cifras, signos de puntuación y letras (mayúsculas y minúsculas) con poquísimas probabilidades de error. Un error típico consiste en confundir un carácter con otro, por ejemplo, «5» con «S», pero, normalmente, menos de un carácter de cada 100.000 resulta erróneo por este motivo.
Otro tipo de inconveniente, que puede ocurrir una vez por cada 10.000 caracteres, es que la máquina no logre identificar un signo.
Si una máquina sólo tiene que identificar números, por ejemplo en demandas de bonificaciones de seguros, no es necesario entonces que la calidad de escritura sea tan buena como cuando hay que identificar números y letras simultáneamente. Antes de reconocer la forma de un carácter, una máquina lectora explora el texto mediante una fila de células fotoeléctricas, o bien mediante un dispositivo láser o, menos frecuentemente, con una telecámara. Tras haber sido analizado, un carácter es representado por la máquina como una matriz de bits.

0000000000	*0000000000*	*0000000000*
0011111000	*0011111000*	*0011111000*
0111111100	*0111111100*	*0111111100*
0110001110	*0110001110*	*0110001110*
0000000110	*0000000110*	*0000000110*
0000000110	*0000000110*	*0000000110*
0000001100	*0000001100*	*0000001100*
0000011000	*0000011000*	*0000011000*
0000110000	*0000110000*	*0000110000*
0011100000	*0011100000*	*0011100000*
0111111110	*0111111110*	*0111111110*
0111111110	*0111111110*	*0111111110*
0000000000	*0000000000*	*0000000000*

En el diagrama anterior, que representa el número 222, el «1» corresponde al negro y el «0» al blanco. El hombre comprende en seguida que ese código representa el número 2, pero una máquina, para llegar a esta conclusión, ha de efectuar ciertos cálculos.
Uno de los métodos utilizados por las máquinas consiste en comparar esta matriz de 140 bits con otras matrices de 140 bits almacenadas en la máquina, cada una con su propio código. Cuando compara dos modelos, la máquina cuenta el número de bits correspondientes que difieren entre sí. La identificación consiste en encontrar el código del modelo concreto para el cual el número de bits discordantes es mínimo. Este sencillo esquema en la práctica es bastante más sofisticado, pero aun así resulta poco eficiente. En efecto, con demasiada facilidad puede suministrar un código erróneo si la forma del carácter está ligeramente deformada, o si alguno de los «0» o de los «1» es accidentalmente distinto a causa de una impresión imperfecta, manchas en el documento o algún defecto del dispositivo electrónico de exploración.
Para evitar estos inconvenientes se ha puesto a punto una técnica de reconocimiento distinta, basada en la utilización de relaciones lógicas, que permite restablecer el vapor de un bit «ensuciado». Por ejemplo, consideramos la siguiente matriz de bits, en la que algunos bits significativos han sido designados con las letras del alfabeto:

0000000000	*0000000000*	*0000000000*
000aabb000	*000aabb000*	*000aabb000*
00aeeffb00	*00aeeffb00*	*00aeeffb00*
0ae0000fb0	*0ae0000fb0*	*0ae0000fb0*
0ae0ii0fb0	*0ae0ii0fb0*	*0ae0ii0fb0*
0ae0ii0fb0	*0ae0ii0fb0*	*0ae0ii0fb0*
0ae0000fb0	*0ae0000fb0*	*0ae0000fb0*
0cg0000hd0	*0cg0000hd0*	*0cg0000hd0*
0cg0jj0hd0	*0cg0jj0hd0*	*0cg0jj0hd0*
0cg0jj0hd0	*0cg0jj0hd0*	*0cg0jj0hd0*
0cg0000hd0	*0cg0000hd0*	*0cg0000hd0*
00cgghhd00	*00cgghhd00*	*00cgghhd00*
000cc0d000	*000cc0d000*	*000cc0d000*
0000000000	*0000000000*	*0000000000*

El carácter considerado será identificado como 0 si:
(al menos cinco «**a**» son «**1**»
o
al menos cuatro «**e**» son «**1**»)



BEI Litton

El microordenador se ha introducido en las cajas registradoras. Estas máquinas contienen un pequeño sistema completo, con CPU, memorias e interfaces I/O.

y
(al menos cuatro «**b**» son «**1**»
o
al menos cuatro «**f**» son «**1**»)
y
(al menos cinco «**c**» son «**1**»
o
al menos cuatro «**g**» son «**1**»)
y
(al menos cinco «**d**» son «**1**»
o
al menos cuatro «**h**» son «**1**»)
y
(al menos tres «**i**» son «**0**»)
y
(al menos tres «**j**» son «**0**»)

Algunas de las máquinas más costosas pueden reconocer las letras mayúsculas escritas a mano en letra de imprenta, con tal de que haya intervalos adecuados entre letras contiguas.

Antes de identificar un carácter escrito a mano, la máquina ha de determinar en el documento una región que contenga sólo dicho carácter, excluyendo las partes pertenecientes a caracteres contiguos. Esta operación es fácil sólo si entre caracteres contiguos se han dejado intervalos. La proliferación de ordenadores basados en microprocesadores ha provocado una avalancha de demandas para la conversión de datos numéricos o textos en representaciones codificadas utilizables por el ordenador. Ello significa que la presión del mercado para la fabricación de máquinas lectoras nunca ha sido tan grande. Por otra parte el número de patentes de máquinas lectoras concedido anualmente ha disminuido notablemente a finales de los años setenta. El mercado mundial de máquinas lectoras está dominado por un reducido número de empresas norteamericanas y japonesas, que en ocasiones han sufrido enormes pérdidas financieras al fabricar máquinas que ha sido imposible vender a causa de su elevado costo y de la limitada capacidad de absorción por parte de los usuarios.

La relación entre amplitud de mercado y costos se está equilibrando cada vez más en los últimos años, debido tanto a la demanda creciente como a los progresos de la electrónica.

Diagramación

La diagramación es la operación mediante la cual se representa un programa con una serie de símbolos gráficos: el diagrama resultante es la carta de flujo (flow-chart) del programa. Cada función que un ordenador puede realizar tiene su correspondiente símbolo gráfico; por lo tanto, una vez que el diagrama de flujo haya sido correctamente preparado, la escritura del programa se reduce a una tarea de traducción de su contenido en las correspondientes instrucciones del lenguaje de programación elegido.

Los símbolos gráficos utilizables se dividen en tres grupos fundamentales:

- símbolos relativos a las elaboraciones (cálculos, decisiones, etc.)
- símbolos de las operaciones de entrada y salida de datos
- símbolos relativos a la memorización

En el gráfico inferior se muestran los símbolos generales más utilizados, aunque hay que advertir que en la programación de mini y microordenadores el número de símbolos usados es mucho más restringido. Por lo tanto, sólo se describirán los símbolos más comunes, que usaremos corrientemente en nuestras aplicaciones, mientras que de los otros (poco o nada usados) sólo se dará una somera descripción.

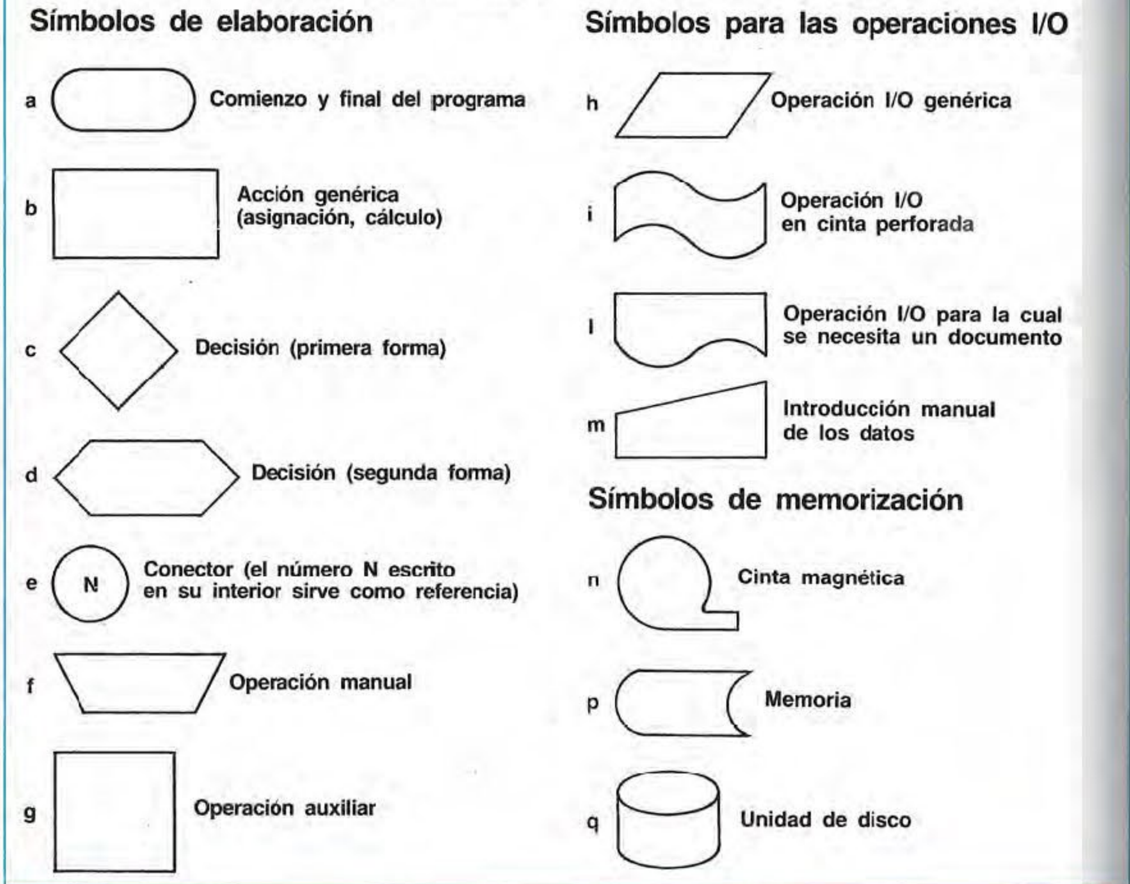
Símbolos de uso corriente

Los símbolos de elaboración son los más usados, ya que las operaciones de elaboración son el cuerpo principal de todo programa.

El símbolo **a** del gráfico inferior se utiliza para indicar el comienzo y el final de un programa. Cualquier programa, por complejo que sea, tiene la forma general que se muestra en el gráfico superior de la página contigua. Las flechas indican el sentido según el cual se procede en la elaboración.

El símbolo **b**, completado por las pertinentes indicaciones, representa una elaboración genérica, por ejemplo, un cálculo.

SIMBOLOGIA DE LOS DIAGRAMAS DE FLUJO





representa una elaboración genérica, por ejemplo, un cálculo.
En el procedimiento de gestión de un almacén antes considerado, el Costo Total se obtiene como producto de los valores Cantidad (número de artículos) y Costo Unitario; esta operación tendrá que ser indicada con el símbolo **b** de la manera siguiente:

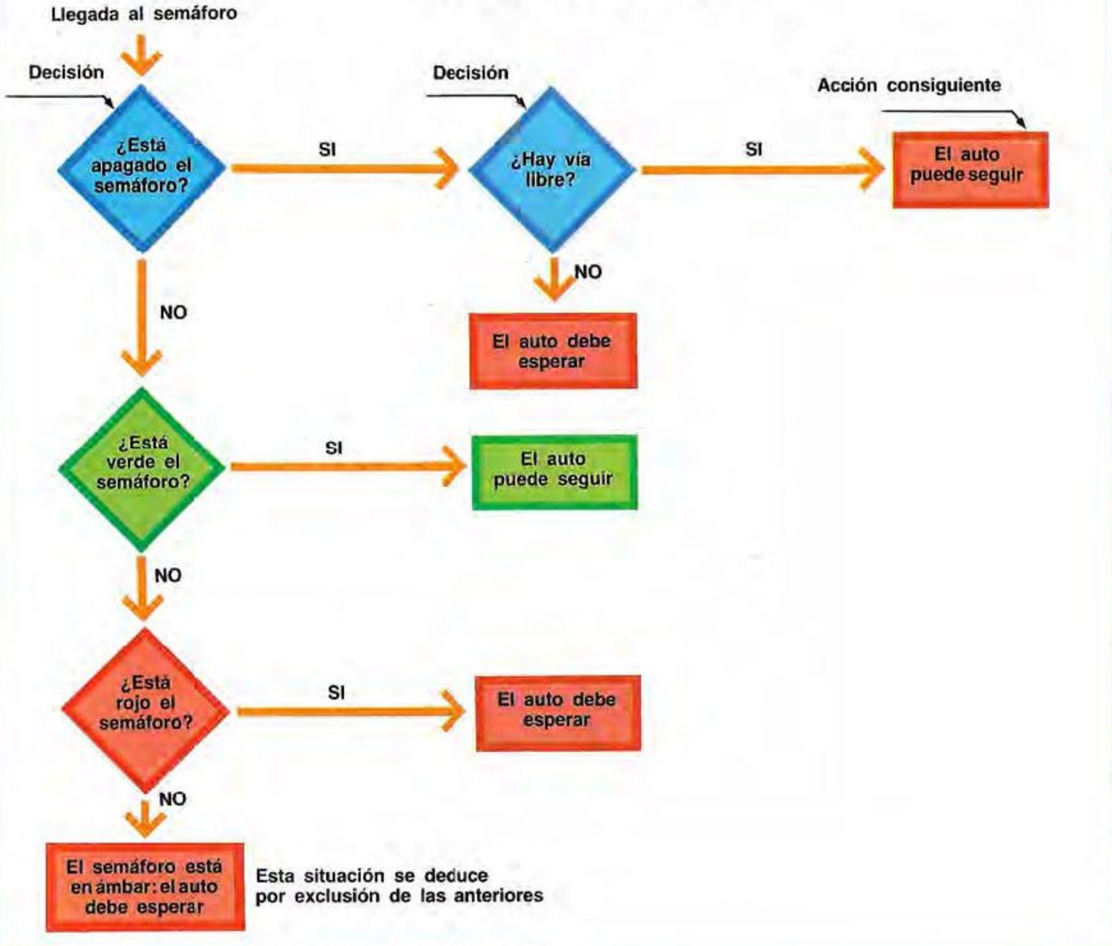
Costo Total =
Cantidad ×
Costo Unitario

Los símbolos **c** y **d** son equivalentes, y ambos representan una decisión.
La utilización de uno u otro depende sólo de la longitud del comentario que hay que escribir en su interior.
La operación de decisión es una operación compuesta que se desarrolla en dos fases:

- primera fase: comparación (entre dos o más valores)
- segunda fase: elección de una de las posibles vías a seguir en función del resultado de la comparación

Consiceremos, por ejemplo, un automóvil que

ESQUEMA LOGICO DEL COMPORTAMIENTO DE UN AUTOMOVIL ANTE UN SEMAFORO

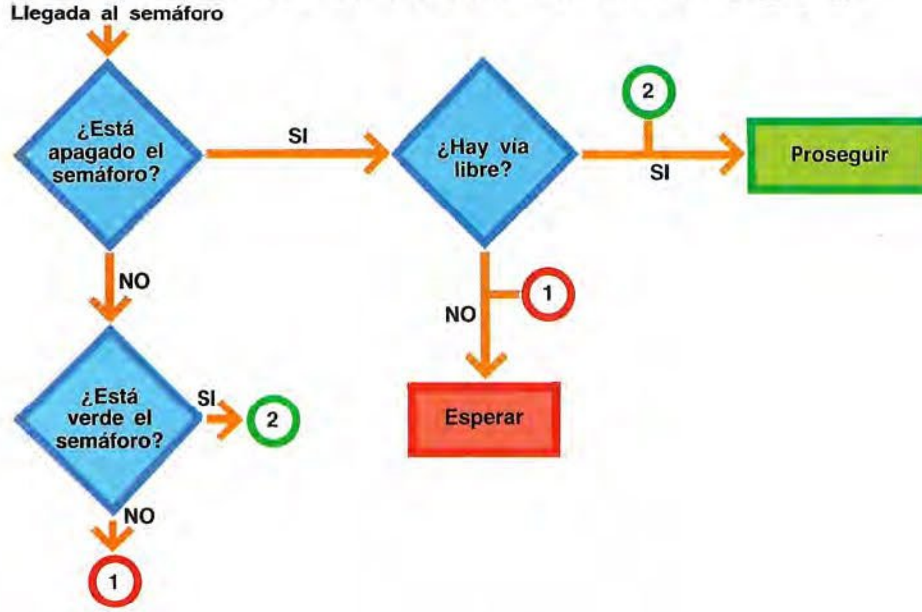


llega a un semáforo. Todos sus comportamientos posibles se pueden esquematizar utilizando los símbolos **b** y **c** (ver gráfico de pág. 169).El símbolo **e** (conector) del gráfico de la pág. 168 se utiliza para indicar el punto del diagrama de flujo desde el que ha de proseguir la elaboración del programa (tras la ejecución de las indicaciones suministradas por el bloque anterior) sin dibujar toda la línea de conexión (línea de flujo). Sirve principalmente para facilitar la lectura de los diagramas o para indicar la continuidad de un diagrama complejo que ocupe varias páginas. Con su empleo, el diagrama toma la forma indicada en el gráfico adjunto.

Símbolos gráficos menos frecuentes

El símbolo **f** indica una operación manual genérica, por ejemplo, la sustitución de un diskette o del tipo de papel que alimenta la impresora. También en este caso el símbolo ha de ir acompañado de una breve descripción de las funciones. Como es una operación manual, el programa ha de prever una pausa que dé al operador tiempo para cumplir el requisito y que se obtiene previendo en el programa la prosecución sujeta al consenso explícito del operador.
Al llegar al punto de la operación manual, el programa emite un requerimiento de consenso y permanece a la espera de la respuesta; el operador tiene tiempo de efectuar la operación manual y, cuando está listo, introduce la orden de continuar. En el gráfico superior de la pág. 171 se muestra el diagrama de flujo de primer nivel que describe este procedimiento. El símbolo **g** indica una operación auxiliar cualquiera, que no concierne estrictamente al programa, mientras que el símbolo **i** indica la emisión o la lectura de una cinta perforada. Esta forma de registro de datos, basada en la perforación del valor del dato en una cinta de papel (a cada orificio le corresponde un nivel 1; ver gráfico inferior de pág. 171), ya está anticuada, pues ha sido sustituido por la grabación magnética.

SIMPLIFICACION DEL DIAGRAMA DE FLUJO ANTERIOR CON EL USO DE LOS SIMBOLOS DE CONJUNCION ① y ②



Aquí la pregunta «¿está rojo el semáforo?» puede omitirse, pues cualquiera que sea el resultado, la acción que sigue es siempre «espera».
En cada programa hay que suministrar datos a la máquina para recuperarlos elaborados. La introducción de datos es una operación de entrada (Input); la devolución por parte de la máquina de los datos elaborados (por ejemplo, una impresión) es una operación de salida (Output).
El signo gráfico más utilizado para indicar una operación de I/O genérica es el símbolo **h** (ver gráfico de pág. 168). El tipo de operación de que se trata realmente (entrada o salida) hay que escribirlo en su interior.

METODO DE ESPERA PARA LAS OPERACIONES MANUALES



En este punto se emite la demanda. Por ejemplo: cambiar el papel de la impresora; una vez listo introducir 1

El ordenador permanece a la espera de la respuesta

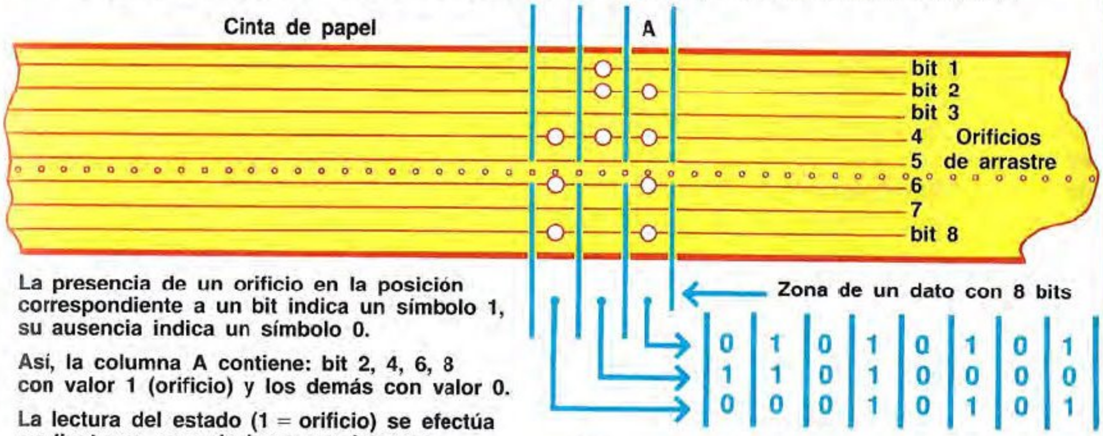
El operador, al terminar el cambio de papel, introduce el valor 1. Cualquier otro valor provoca una nueva espera

El diagrama de flujo detallado puede sustituirse por un solo símbolo de operación manual:



Esta decisión equivale a: ¿el valor introducido es 1?

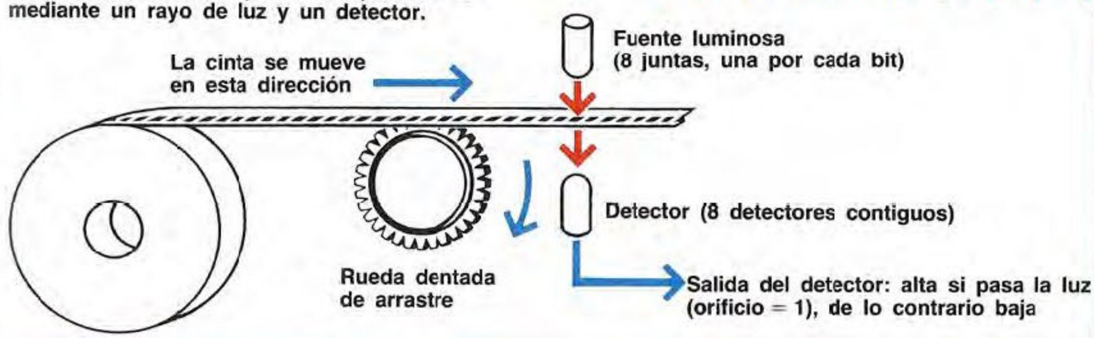
EJEMPLO DE MEMORIZACION DE DATOS EN CINTA PERFORADA



La presencia de un orificio en la posición correspondiente a un bit indica un símbolo 1, su ausencia indica un símbolo 0.

Así, la columna A contiene: bit 2, 4, 6, 8 con valor 1 (orificio) y los demás con valor 0.

La lectura del estado (1 = orificio) se efectúa mediante un rayo de luz y un detector.





Olivetti

Uso del ordenador para proyectar: proporciona datos exactos, ideas nuevas y ahorra tiempo.

El símbolo **l** indica una operación I/O para la cual se requiere un documento. Se utiliza principalmente para llamar la atención sobre las operaciones de impresión que requieren una modulación especial; por ejemplo, la impresión de hojas de salarios.

El símbolo **m** indica genéricamente una introducción manual de datos. La lógica a seguir es la misma que para el símbolo **f** (operación manual genérica). El símbolo **n** indica la cinta magnética. Este tipo de soporte ha reemplazado casi por completo a la cinta perforada. Se utiliza para memorizar los datos de uso menos frecuente (para los que requieren acceso frecuente es preferible el diskette). Con el símbolo **p** se indican la memoria y las operaciones relativas a ella; el símbolo **q** se refiere a los discos, es decir, a las memorias de masa más usadas.

Ejemplo de aplicación

Lo que hemos visto hasta ahora sobre la diagramación nos permite plantear y resolver de manera lógica un problema determinado. El uso de la diagramación será ampliamente ilustrado en lo sucesivo, pero ya desde ahora es posible poner un ejemplo completo de cómo hay que proceder para trazar un diagrama de flujo.

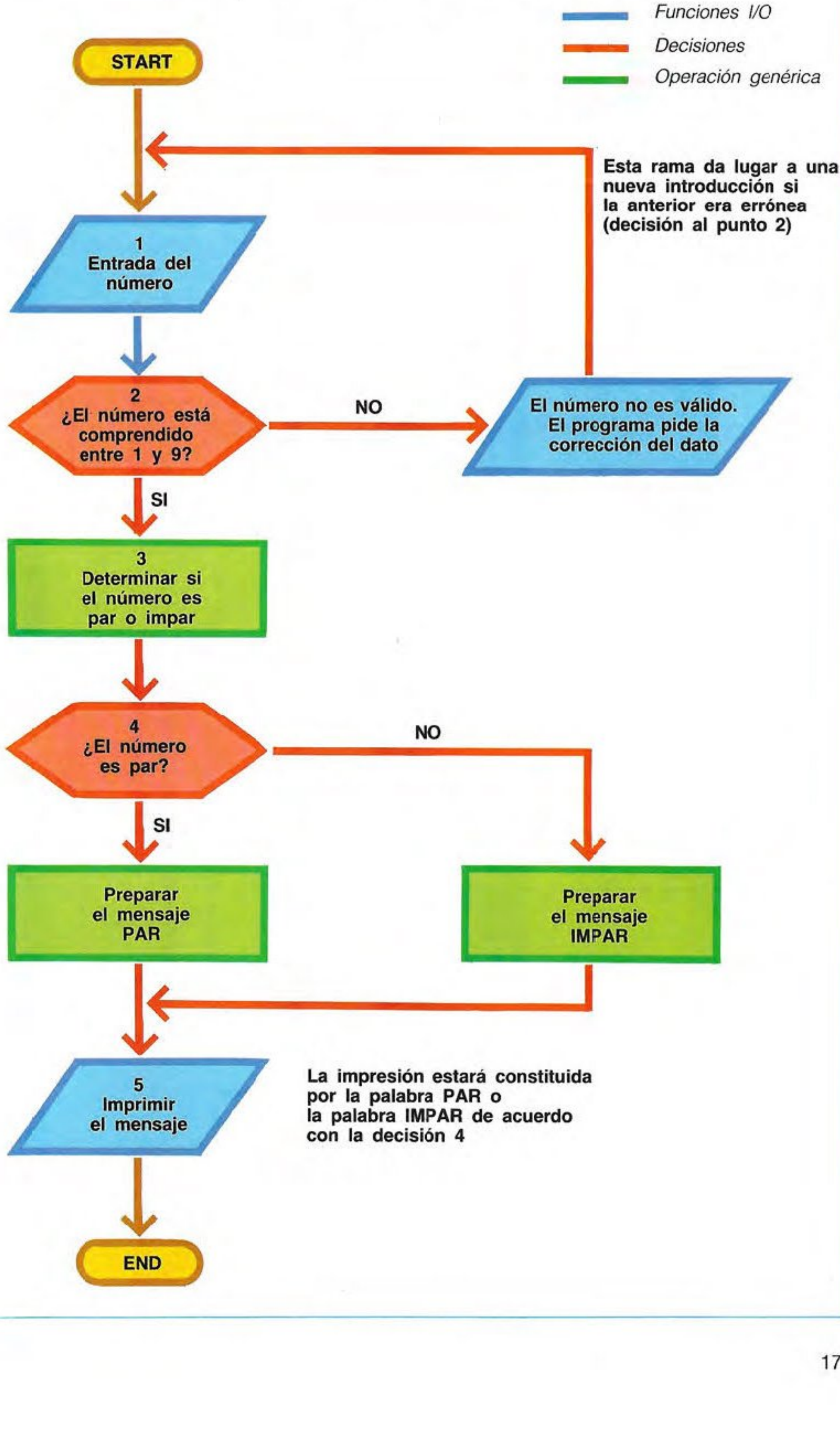
Supongamos que hay que resolver el siguiente problema: leer un número comprendido entre 1 y 9 y escribir si es par o impar.

El primer paso consiste en dividir el problema principal en un determinado número de problemas más simples, que en este caso podrían ser los siguientes:

1 / lectura del número (operación I/O)
2 / controlar que esté comprendido entre 1 y 9 (decisión)
3 / determinar si el número es par o impar (acción genérica)
4 / elección de la respuesta a dar (decisión)
5 / impresión de la respuesta (operación I/O)

Al traducirlo en símbolos se obtiene el diagrama

DIAGRAMA DE FLUJO DE PRIMER NIVEL QUE RESUELVE EL PROBLEMA EJEMPLIFICADO



de flujo de primer nivel mostrado en el gráfico de la pág. 173.

Si se desea aplicar la técnica HIPO, el esquema del programa anterior se convierte en:

— entrada: un número
— elaboración: control de la validez del número (comprendido entre 1 y 9) y determinación de si el número es par o impar
— salida: si el número no está comprendido entre 1 y 9, mensaje de error; de lo contrario, mensaje PAR o IMPAR

Los bucles

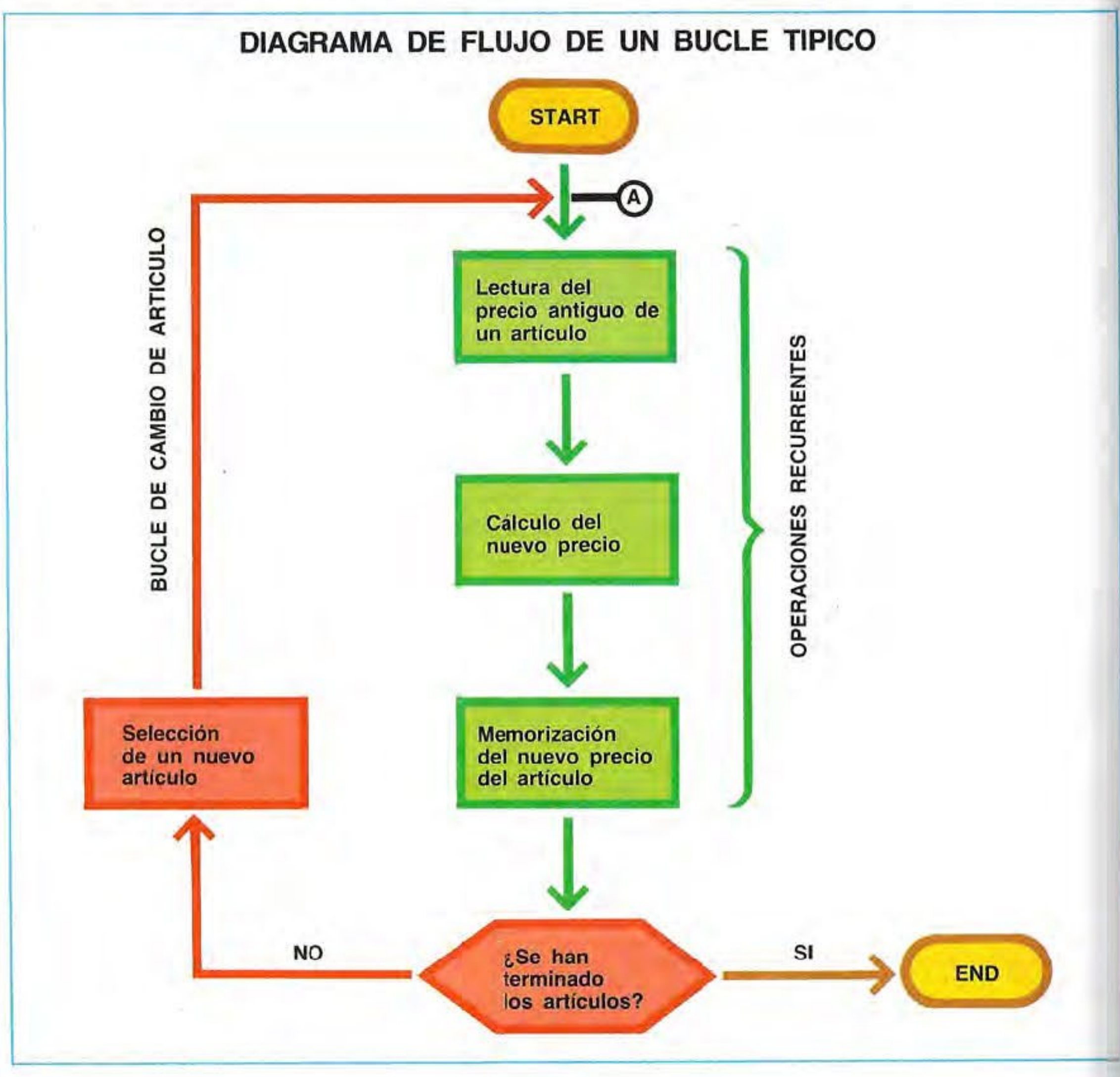
El uso de un sistema de microordenador resulta muy útil al efectuar cálculos recurrentes.

Esta situación se presenta cuando un mismo cálculo (o serie de cálculos) se ha de efectuar con distintos datos. La situación de cálculo recurrente se resuelve mediante el empleo de un **bucle** (loop), es decir, un programa (o parte de programa) que al final de un cálculo vuelva al propio comienzo, tome un nuevo dato y repita con él el mismo cálculo. Este ciclo (bucle) ha de repetirse hasta que se agoten todos los datos.

Por ejemplo, si queremos conocer los nuevos precios de los artículos de un almacén tras un aumento del 12%, hay que efectuar para cada artículo el siguiente cálculo recurrente:

$$\text{nuevo precio} = \text{precio anterior} + \left(\frac{\text{precio anterior}}{100} \times 12 \right)$$

En el gráfico inferior se muestra el diagrama de flujo de primer nivel que realiza el cálculo, en el cual las partes que constituyen el bucle se se-

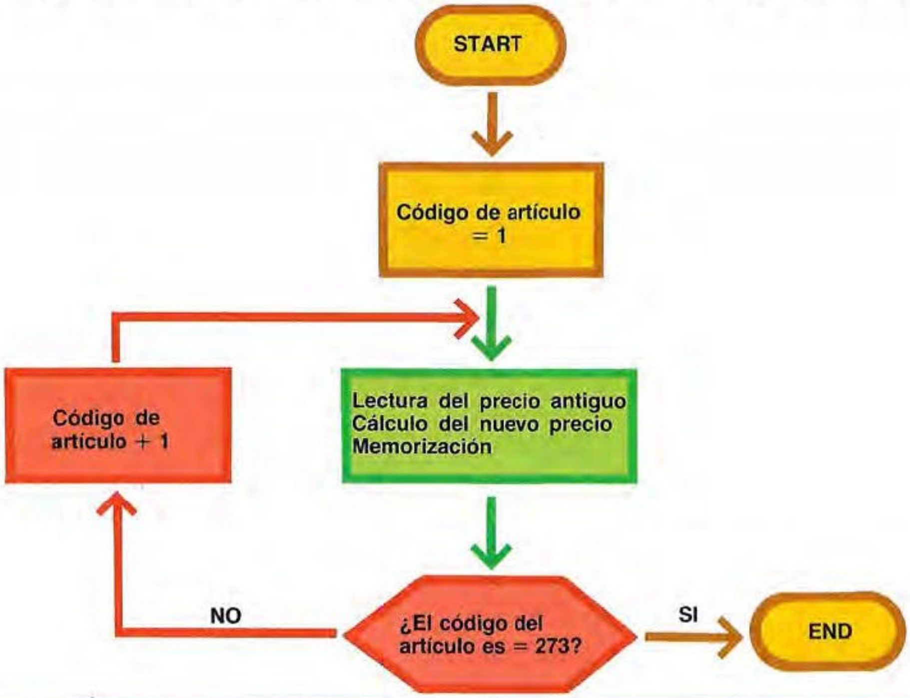


ñalan en rojo. Para cada artículo, el retorno al comienzo de la elaboración está en el punto A.

Bucle entre límites explícitos

Si el almacén contiene 273 artículos diferentes, indicados con números de código progresivos entre 1 y 273, el bucle tendrá que repetirse 273 veces (una por artículo) a partir del artículo número 1. Para saber si los artículos se han terminado basta comprobar si se ha llegado al número 273, mientras que para seleccionar un nuevo artículo basta con sumar 1 al código del artículo anterior. En el gráfico inferior se muestra el diagrama de flujo que tiene en cuenta estas modificaciones. En este programa el código de artículo es un número (del 1 al 273) que indica qué artículo es objeto de elaboración. Dicho número se denomina **índice del bucle**. En general, un

BUCLE MODIFICADO PARA EL CONTROL DE LOS ARTICULOS HASTA EL CODIGO 273



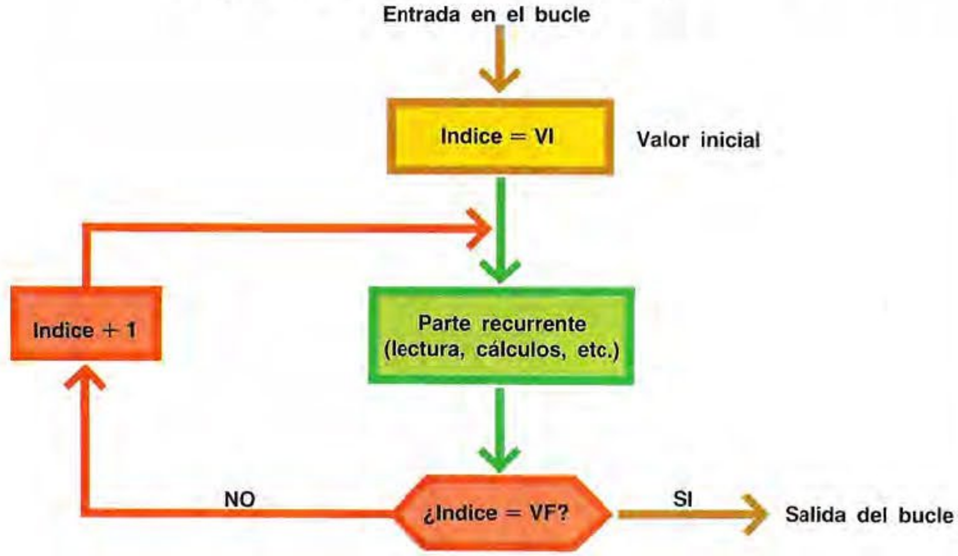
bucle siempre tiene un índice; los valores inicial y final de dichos índices son los límites entre los que se repite el bucle. En el ejemplo anterior, el valor inicial es 1 y el final 273.

Bucle entre límites parametrizados

En el gráfico de esta página, los límites entre los que ha de aplicarse el bucle, es decir, los valores inicial y final del índice (1, 273), se explicitan

de forma numérica. Pero también se puede utilizar una forma indirecta, es decir, dar un nombre simbólico cualquiera a ambos extremos. Naturalmente, antes de iniciar el bucle hay que indicar cuáles son los valores numéricos de los dos nombres simbólicos. Llamando **VI (Valor Inicial)** y **VF (Valor Final)** a los dos límites, el diagrama de flujo de un bucle genérico puede representarse como se ve en el gráfico.
El hecho de iniciar los extremos de un bucle con

BUCLE MODIFICADO PARA EL FUNCIONAMIENTO ENTRE LOS DOS LIMITES VI Y VF



simbología implícita permite «parametrizar» la función. Se dice que una parte del programa está parametrizada cuando no se desarrolla con valores numéricos expresamente indicados, sino que admite en la entrada nombres simbólicos que constituyen los parámetros; antes de hacer ejecutar las instrucciones parametrizadas, hay que declarar el valor numérico de los parámetros a utilizar. La parametrización permite utilizar las mismas instrucciones para aplicaciones con valores numéricos distintos.
En el ejemplo anterior, si se desea efectuar el cálculo para todos los artículos del almacén, los parámetros tendrán los valores:

- **Valor Inicial** = **VI** = 1
- **Valor Final** = **VF** = 273

Si se quisiera efectuar el cálculo sólo para los artículos del número 20 al número 56, habría que insertar, antes del bucle, dos instrucciones de asignación:

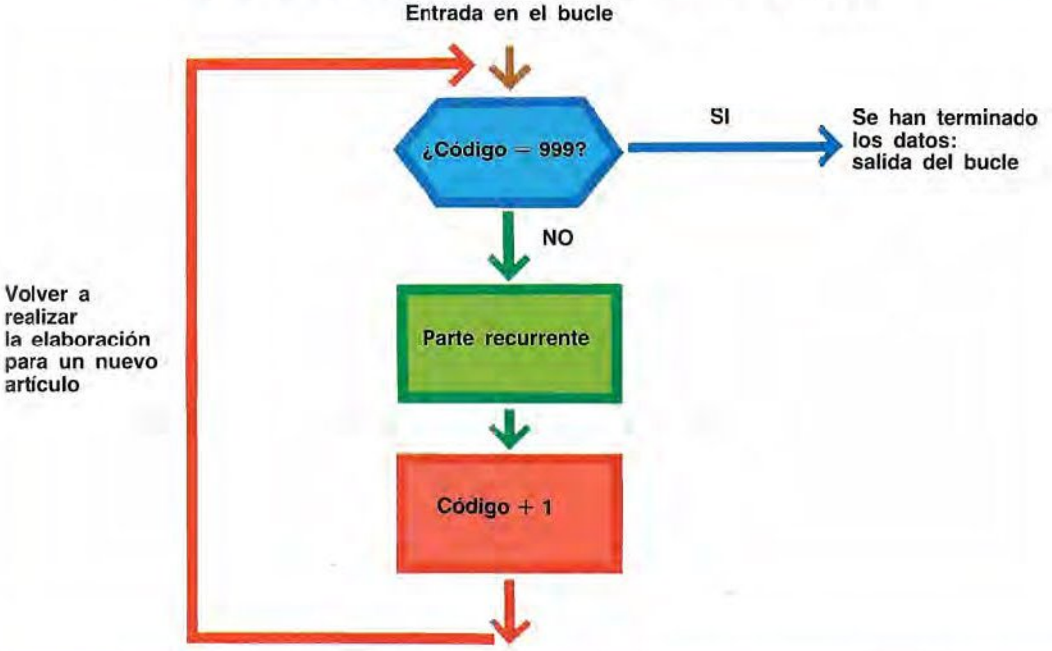
- **VI** = 20 (comienzo en el código 20)
- **VF** = 56 (final en el código 56)

Bucle con indicador de fin de sondeo

Los bucles mencionados, es decir, el de límites explícitos y el de límites parametrizados, terminan sólo cuando todos los datos han sido examinados, pero tienen un defecto: hay que saber antes el número de datos (en el ejemplo, 273). En las aplicaciones, dicho número no siempre es conocido. En tales casos hay que adoptar un tipo de bucle que pueda determinar por sí mismo cuál es el valor final. El método consiste en escribir al final de los datos un valor convencio-

nal fácilmente reconocible; al encontrar dicho valor, el bucle lo reconoce y, por tanto, puede terminar. El valor simbólico de **Fin Datos** se escribe al final de la frase de introducción, y se denomina **Tapón**.
Volviendo al ejemplo del almacén, para señalar el fin de los datos podemos introducir un código de mercancía que no existe: el hallazgo de dicho código significa que los datos se han terminado y se puede salir del bucle.

BUCLE CON INDICADOR DE FINAL DE SONDEO



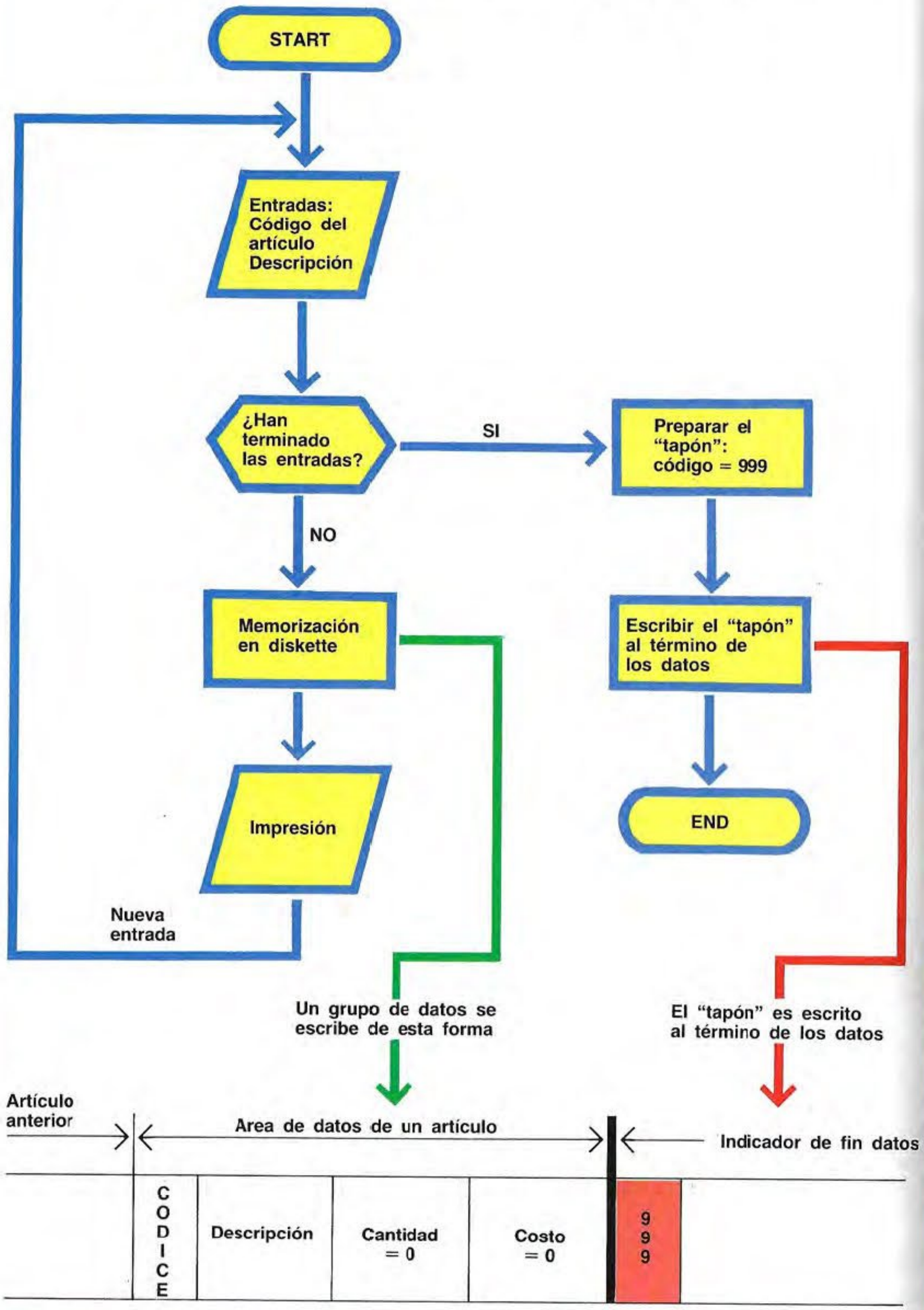
Si el código ficticio es 999, el diagrama de flujo toma la forma del gráfico. En este caso la decisión de salir o no del bucle se toma antes de la **Parte Recurrente**, de lo contrario se efectuarían cálculos con valores erróneos, ya que pertenecen a un código ficticio usado como tapón. Éste, como hemos visto, ha de escribirse en la fase de introducción de datos, y ha de tener un valor que no pertenezca a los mismos. En el ejemplo del almacén, hay que prever dicho valor en el programa de creación de las denominaciones de almacén. En este caso, el diagrama de flujo se modifica (ver gráfico de pág. 178).

Bucle con ruptura de código

Esta clase de bucle permite el desarrollo de las funciones recurrentes mientras un determinado parámetro conserva el valor inicial; cuando el parámetro cambia de valor se dice que hay una **Ruptura de Código** y el bucle se interrumpe.
Esta denominación se usa normalmente en los lenguajes Cobol y RPG*. En el Basic, aunque los bucles con ruptura de código también se usan, no tienen un nombre específico.
La técnica de la ruptura de código es muy útil en las elaboraciones selectivas, es decir, en aquellas que han de operar sólo con determinados datos: los que tienen el mismo valor que el código. El código puede ser una parte cualquiera del dato; por ejemplo, en la gestión del almacén se puede tomar como código la cantidad

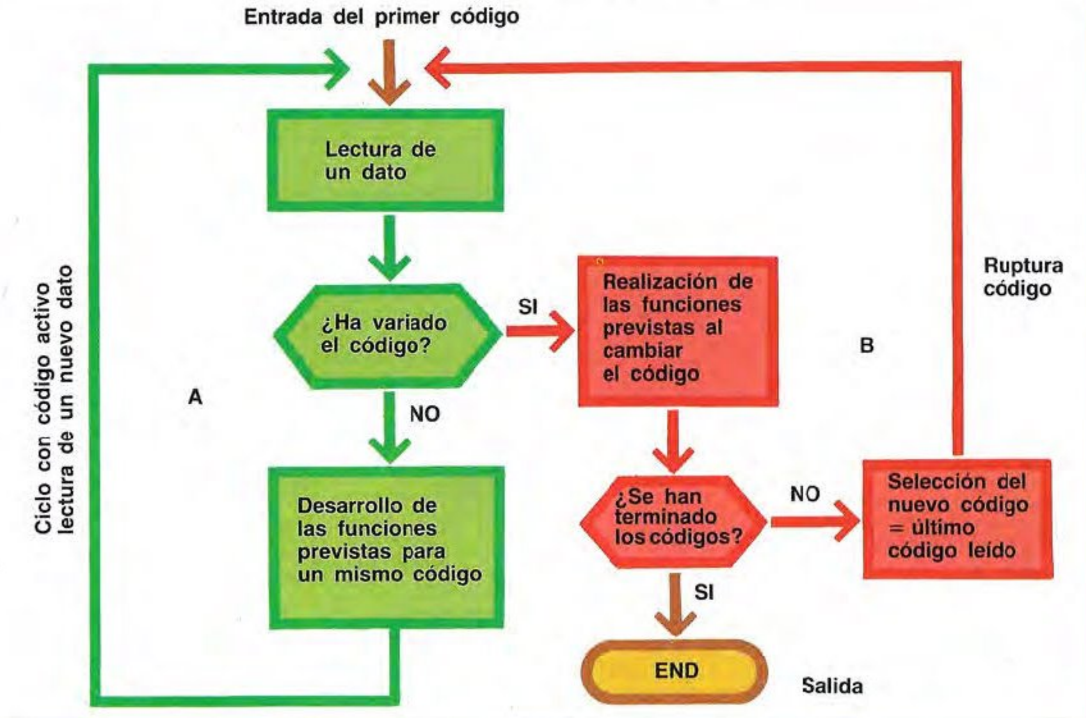
*El lenguaje RPG es muy similar al Cobol, menos complejo y usado sólo en ciertos tipos de ordenador (IBM).

CREACION DE LAS DENOMINACIONES DE ALMACEN



de existencias; en tal caso, las elaboraciones comprendidas en el bucle se efectuarán sólo para los artículos de los que haya las mismas existencias. El diagrama de flujo de primer nivel de un bucle con ruptura de código se muestra en el gráfico de esta página. En él se evidencian dos líneas de flujo: la constituida por las operaciones a realizar con los datos que tienen el mismo código (bloque funcional A) y la constituida por las operaciones a realizar en caso de ruptura del código (bloque funcional B).

DIAGRAMA DE FLUJO DE PRIMER NIVEL GENERICO DE UN BUCLE CON RUPTURA DE CODIGO



De haber ruptura de código tras haber realizado las funciones requeridas, se controlará si los códigos posibles se han terminado. En caso afirmativo, el programa finaliza; si no, se carga un nuevo código y se vuelve a entrar en el bucle. Un significativo ejemplo del uso de esta técnica lo constituye el análisis de las ventas efectuadas por un grupo de representantes. Al final de un determinado período de tiempo se desea conocer el importe de las ventas para cada uno de los representantes; en este caso, el código es el número del representante, y la función a realizar

La revolución de la imprenta

La fotocomposición fue acogida como la mayor revolución en el sector de la imprenta después de la invención de los caracteres móviles. Los sistemas electrónicos de reproducción, complementados por las transmisiones vía satélite, permiten publicar simultáneamente en todo el mundo las ediciones de diarios y revistas. En lugar de grandes establecimientos de prensa con numerosos medios de transporte para distribuir la información relativa a toda una nación, la prensa puede ser de dimensiones locales, con sede en establecimientos automatizados de tamaño reducido, con pocas personas encargadas de la supervisión de las distintas operaciones y de los envíos. Algunos semanarios de nivel internacional ya envían páginas completas en películas, listas para imprimir, a diversos lugares del mundo. Uno de los más importantes diarios estadounidenses ha iniciado la transmisión vía satélite de copias de la edición principal. Pero estos son sólo dos ejemplos de una revolución que ha afectado a todas las formas de impresión, desde los periódicos a los billetes de banco.

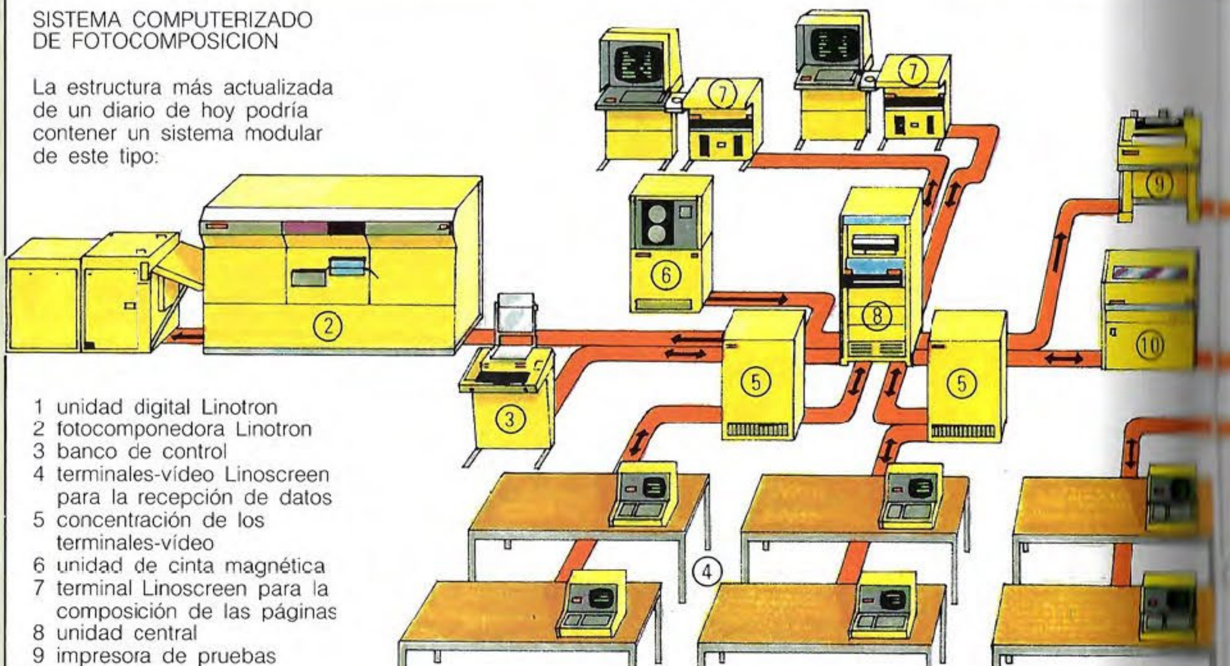
Los diarios y revistas tienen una vida breve y requieren una publicación rápida; por eso han sido de los primeros en adoptar la moderna tecnología electrónica. En el mundo occidental, a excepción de las grandes ciudades, los periódicos abastecen a pequeñas comunidades en las que, en la práctica, no compiten entre sí, por lo que pueden aprovechar mejor las nuevas tecnologías. Las revistas de actualidad, por su parte, adoptan las tecnologías más recientes para mantener su nivel competitivo. Además, como sucede con los más importantes diarios financieros, han de hacer llegar rápidamente los ejemplares a lectores de muchos países de todo el mundo. Por ello recurren a la teletransmisión de copias, para asegurar su difusión en localidades diferentes casi al mismo tiempo, a la vez que utilizan el transporte aéreo para la distribución regional.

Publicidad especializada, información bursátil, guías de los locales de diversión y datos enciclopédicos se prestan a la divulgación electrónica mediante sistemas de Teletexto. El servicio prestado por las pantallas visualizadoras en las oficinas de las líneas aéreas constituye un impresionante ejemplo de las ventajas de la utilización del Teletexto con respecto a los tradicionales listados impresos.

Desgraciadamente, el éxito comercial de los sistemas de Teletexto no es aún tan grande como para pensar en la difusión de los periódicos de

SISTEMA COMPUTERIZADO DE FOTOCOMPOSICION

La estructura más actualizada de un diario de hoy podría contener un sistema modular de este tipo:



1 unidad digital Linotron
2 fotocomponedora Linotron
3 banco de control
4 terminales-vídeo Linoscreen para la recepción de datos
5 concentración de los terminales-vídeo
6 unidad de cinta magnética
7 terminal Linoscreen para la composición de las páginas
8 unidad central
9 impresora de pruebas
10 unidad de discos para almacenar textos y programas

J.Gower

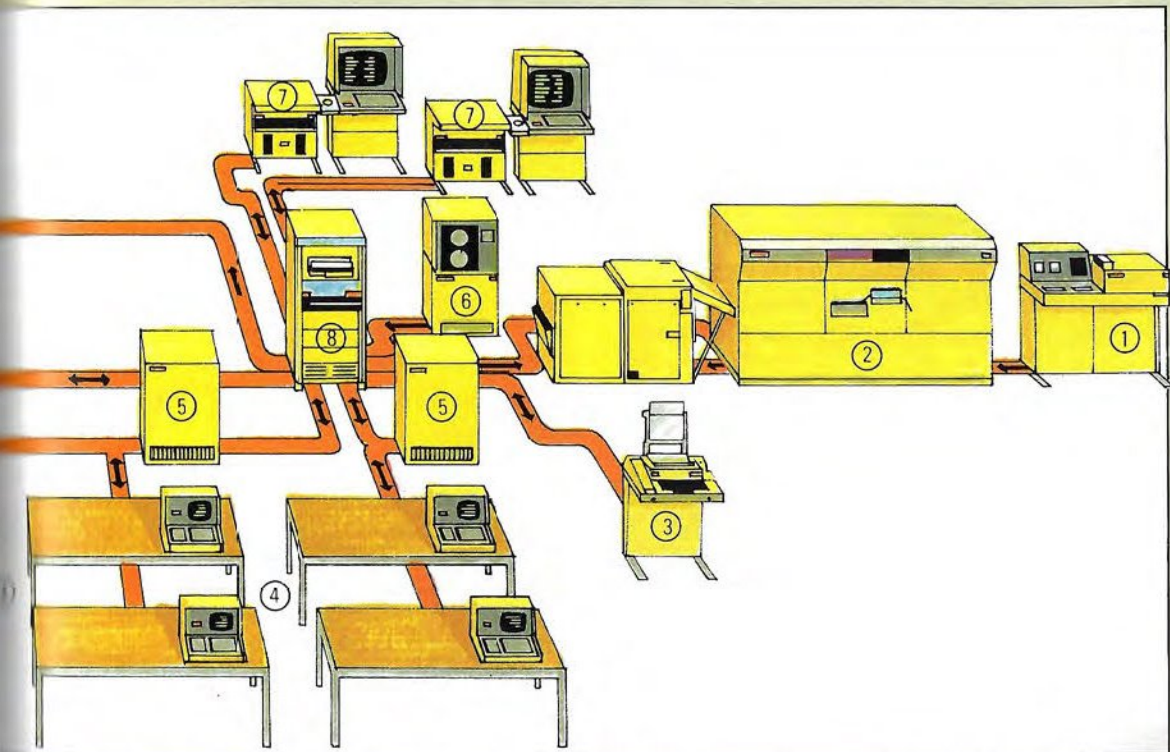
tipo telemático que cabía prever. Algunos expertos consideran que no es que la gente se resista al cambio, sino que, simplemente, el ritmo del cambio es demasiado rápido para que la gente pueda adaptarse a él.

Durante 442 años, el arte de la imprenta, aunque se haya mecanizado, no ha cambiado sus principios. La composición se efectuaba a partir de letras individuales, o bien (desde finales del siglo XIX) líneas enteras, o bloques que formaban palabras o frases completas, todo ello en aleación de plomo. Los impresores de carteles utilizaban la composición en madera para grandes dimensiones. Pero el metal ofrecía la ventaja de que se podía fundir y volver a utilizar.

En su imprenta, el diario londinense Times revolucionó el arte de transferir un texto al papel introduciendo la primera máquina rotativa del mundo. Mientras las imprentas utilizaban antes planchas planas con el material que había que imprimir, las rotativas utilizaban planchas curvas arrolladas alrededor de un cilindro. Hoy día, los sistemas mecánicos de composición, reproducción e impresión están a punto de ser reemplazados por los sistemas de tecnología avanzada de la era electrónica. Actualmente, la mayor parte de la composición se efectúa fotográficamente. Mediante un teclado y una unidad de visualización, un operador teclea el texto, que es manipulado automáticamente por un microprocesador para ser almacenado en cinta o disco magnético. El microprocesador se encarga del espaciado entre las letras y las palabras, y además dispone los caracteres en columnas formadas por filas de longitud constante. Las señales almacenadas magnéticamente se introducen en una máquina fotocomponedora, que las convierte en texto real sobre película o papel fotográfico.

Las primeras fotocomponedoras no eran sino simples adaptaciones de las máquinas componedoras con caracteres obtenidos por fusión. Hoy día, las fotocomponedoras de la cuarta generación funcionan a velocidades increíbles, hasta el punto de que una sola máquina puede convertir en texto todo lo que sale de los teclados de centenares de operadores. También la reproducción de las imágenes está derivando hacia una tecnología totalmente electrónica.

Actualmente, se estima que el 40-50 por ciento de las reproducciones en color del mundo industrializado es analizado electrónicamente. Los analizadores descomponen la imagen original en color en dos, tres o cuatro selecciones (imágenes compuestas por un único color) sobre película. Cuando se superponen e impri-





Con el sistema Sci-Text, la compaginación se efectúa directamente en la pantalla.

men, las selecciones forman una imagen en varios colores, como la real. La generación electrónica de imágenes y textos en un único proceso integrado (denominado prepress*) ha sido introducida por la industria periodística.*

El acoplamiento de estos dos elementos esenciales en un producto impreso ha sido anunciado a menudo, pero sólo recientemente se ha realizado. Texto y figuras son convertidos en señales electrónicas y, de este modo, introducidos en un ordenador. En él, los datos son manipulados (mediante una pantalla y un teclado) y memorizados; pero pueden ser modificados introduciendo, simplemente, las correcciones mediante el teclado, observando los efectos de los cambios aportados y dando instrucciones al ordenador para que almacene la nueva versión. En el momento de imprimir, las señales guardadas en la memoria de masa del ordenador son transferidas a una película, luego a planchas de imprenta, y finalmente, al papel.

Los procedimientos exclusivamente electrónicos son caros, pero recientemente se ha propuesto un método de reproducción electrónica en colores simplificado, adecuado sobre todo para las empresas pequeñas. En vez de recoger en el archivo del ordenador la totalidad de los datos relativos al texto y las imágenes, reproducciones y texto pueden ser combinados sobre una película básica ya perfectamente correcta, de la cual podrán luego sacarse mediante impresión las páginas completas.

Aunque los ordenadores trabajen a enormes velocidades, requieren una programación y la transformación en código de los datos suministrados. Muchas personas dotadas de gran habilidad prefieren trabajar con película: efectivamente, no sólo es fácil de manejar, sino que suministra una imagen directa de lo que se está haciendo. La impresión rotativa mediante cilindro grabado con ácido ofrece la mejor calidad posible sobre papel ligero y barato. Este es el procedimiento empleado en las revistas y catálogos de gran difusión. Con la adopción de las más recientes técnicas electrónicas, el proceso de grabación de los cilindros se integra con la conversión del original en color directamente en la imagen a imprimir. Dos grandes desventajas del sistema son el elevado costo y la lentitud del proceso de grabación de los cilindros.

La grabación mediante láser se efectúa cubriendo el cilindro con un revestimiento de plástico, sobre el cual un rayo láser, controlado por un analizador de colores u otra unidad, graba la imagen a imprimir. Este proceso promete una reducción de costos; sin embargo, la mayor parte de la producción mediante cilindros utiliza el grabado químico con ácidos controlados electrónicamente.

En la imprenta automatizada Harris, de Estados Unidos, el proceso de impresión es literalmente continuo. Dos unidades imprimen una sección de un libro, en uno o dos colores, mientras otras dos unidades se preparan para imprimir la sección siguiente. Un microprocesador acelera las unidades de relevo hasta conferirles la máxima velocidad, mientras que detiene automáticamente las otras unidades impresoras una vez que se ha producido el número de ejemplares preestablecido. Hoy día, los distintos procedimientos de impresión se superponen. Un ejemplo lo constituye el éxito del sistema de conversión Toppan, de origen japonés, en el que las películas preparadas para la impresión en offset *se convierten en películas para la impresión por fotograbado. Las mismas películas de separación de colores se utilizan para imprimir con uno u otro sistema. Se puede imprimir en* offset *y luego reimprimir en fotograbado, o imprimir en* offset *en un sitio y en fotograbado en otro.*

Dada su flexibilidad de funcionamiento, la industria de la imprenta se halla en buenas condiciones para hacer frente a las actuales demandas de comunicaciones rápidas procedentes de todo el mundo.

SOLUCIONES AL TEST 4

1 / b

2 / a: la instrucción 106 contiene el codigo RETURN, que activa el mecanismo de retorno a la instrucción siguiente a la del salto, por lo que después de la 106 el control vuelve a la instrucción 20.

b: en este caso no está el código RETURN y el programa prosigue serialmente a partir de la instrucción 106, es decir, ejecuta la 107 y siguientes.

3 / b: ALU son las siglas de Arithmetic Logic Unit (Unidad Lógico Aritmética) y, como sugiere el nombre mismo, sirve para la realización de los cálculos.

4 / a

5 / c

6 / El error consiste en el código RETURN en la instrucción 33. Este código sólo se debe utilizar para el retorno después de un salto, y en el programa no hay ningún salto que lleve a la 33. Por el contrario, el RETURN de la instrucción 57 es necesario, ya que hay salto (instrucción 32).

7 / b: el interrupt es un medio para detener la actividad en desarrollo y desviar los recursos del sistema hacia un servicio requerido desde el exterior.

8 / a: falso. Las EPROM contienen programas. No se puede escribir en ellas ni borrarlas si no se utilizan aparatos especiales, luego no se pueden utilizar para memorizar datos.

b: falso. Las unidades de disco pueden contener desde un mínimo de unos 250.000 caracteres hasta varios millones (en los tipos más grandes se puede llegar a centenares de millones). El valor 32.000 es un orden de la capacidad de memoria de una máquina pequeña.

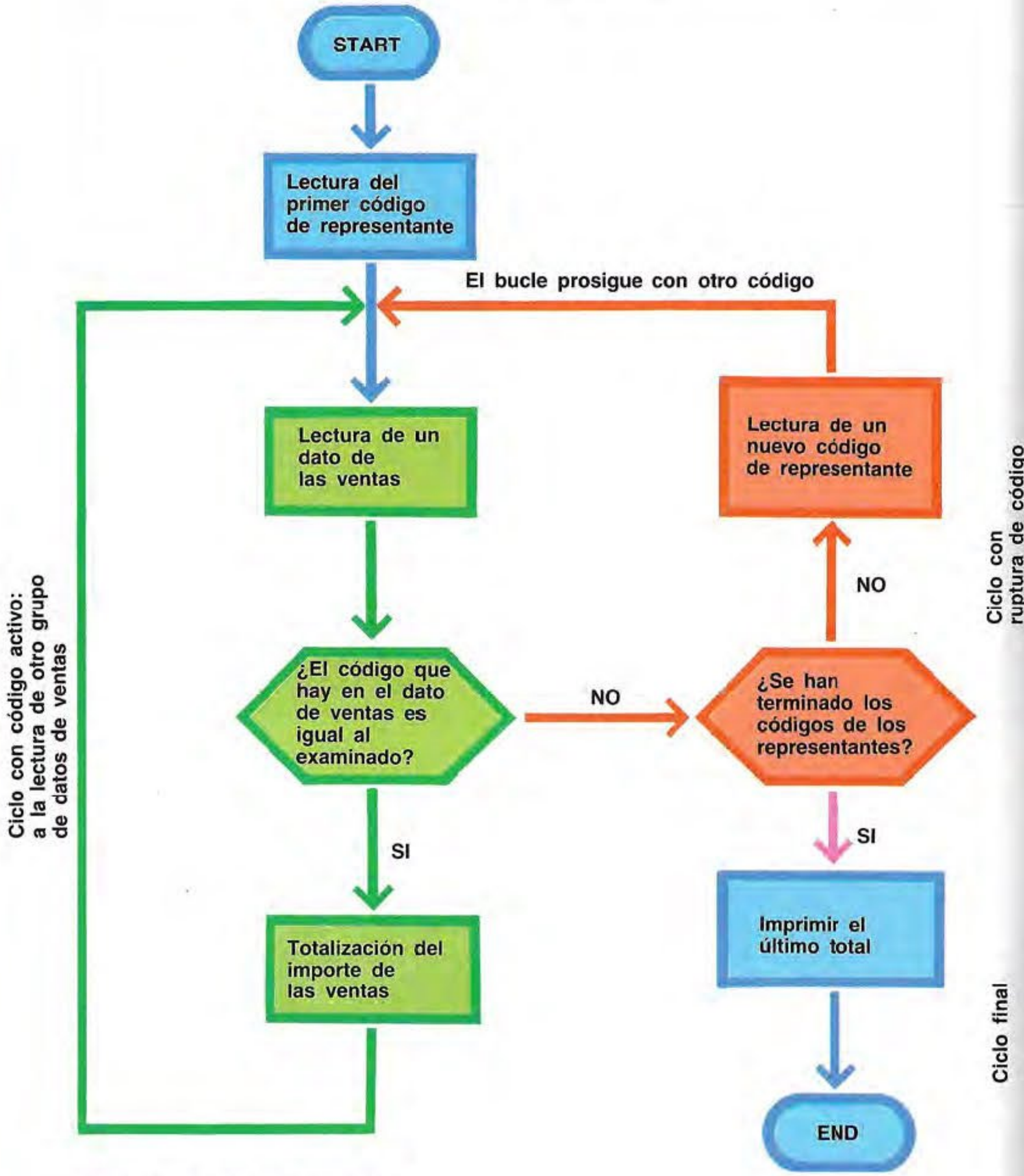
c: cierto. En la ROM se ha de escribir con aparatos especiales.

d: falso. La CPU memoriza lo que está efectuando y se dedica a la nueva actividad. Al final, vuelve al desarrollo normal del programa.

e: la afirmación es cierta a medias. La pantalla normalmente no genera interrupt; por el contrario, para el teclado es normal hacerlo.

f: cierto. Debido a la gran cantidad de datos intercambiados con este periférico, el DMA es la única manera de acelerar los tiempos de transferencia.

DIAGRAMA DE FLUJO DE UN BUCLE CON RUPTURA DE CODIGO: EJEMPLO DE APLICACION



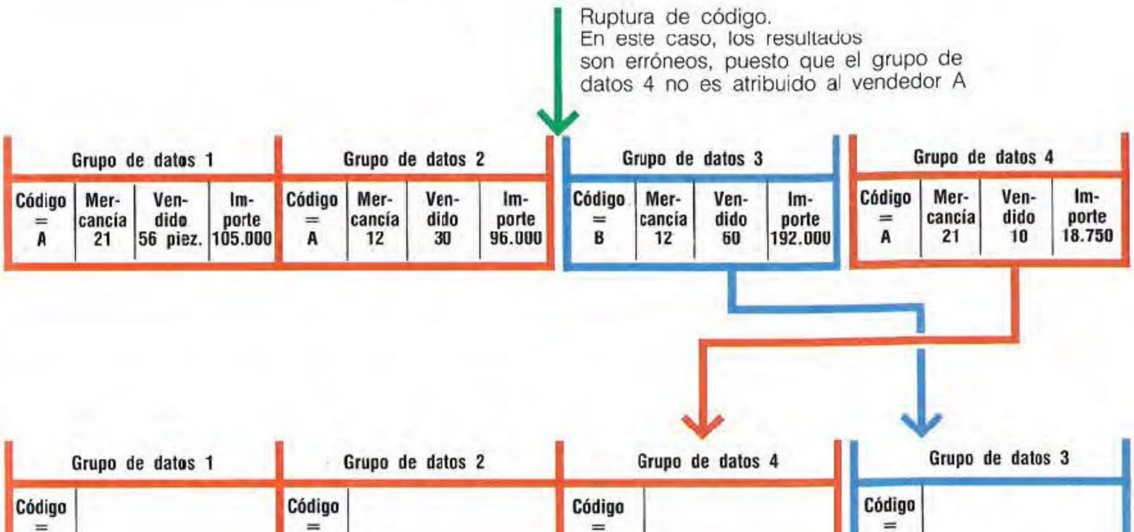
Formato de los datos de ventas

	Código del Represen-tante	Código de la Mercancía	Cantidad Vendida	Importe	Código del Represen-tante	Código de la Mercancía	

⟵ Grupo de datos de una venta ⟶

con el código activo es la suma de las ventas hechas; la función a efectuar con ruptura de código es la impresión del total. En el gráfico de la pág. 184 se muestran el diagrama de flujo que resuelve el problema y una hipótesis sobre la forma que pueden adoptar los datos de las ventas. Con el método de la ruptura de código, los datos han de ordenarse por códigos crecientes o decrecientes. Supongamos que los representantes sólo sean dos, codificados respectivamente con A y B, y que los datos de las ventas hayan sido introducidos sin orden, como se indica en el gráfico inferior. En este caso, la ruptura del código (del valor A pasa a B) no indica el final de los datos del primer vendedor: la totalización de los importes es errónea, pues no se añade el contenido del grupo de datos 4. Este último dato nunca se considerará, ya que al entrar en el bucle por segunda vez, tras la ruptura del código A, se extraerán sólo los datos de B. Antes de efectuar el bucle es, pues, necesario «reordenar» los datos, disponiéndolos, por ejemplo, por códigos crecientes. En este caso, las posiciones de los grupos 3 y 4 quedarán invertidas, y a la ruptura del código se habrá totalizado todo lo relativo al vendedor A.

RUPTURA DE CODIGO





Operadores manejando terminales de main-frame.

Resumiento, las principales clases de bucle son:

— ENTRE LIMITES EXPLICITOS: los valores iniciales y finales se indican con el propio valor numérico en las instrucciones del bucle.

— ENTRE LIMITES PARAMETRIZADOS: los valores iniciales o finales se indican con nombres simbólicos a los cuales, antes de iniciar el bucle, hay que asignar valores numéricos. Con este método el bucle puede utilizarse en distintos puntos del programa.

— CON INDICADOR DE FIN DE EXPLORACION: el valor inicial está, normalmente, definido, mientras que el final coincide con la identificación de un determinado código en el grupo de datos.

— CON RUPTURA DE CODIGO: el bucle parte del primer grupo de datos y termina con el último. Se desarrolla según dos líneas: con código activo para las funciones a realizar en el ámbito del código mismo; con ruptura, para las funciones de fin de código. Para este tipo de bucle es necesario que los datos estén ordenados.

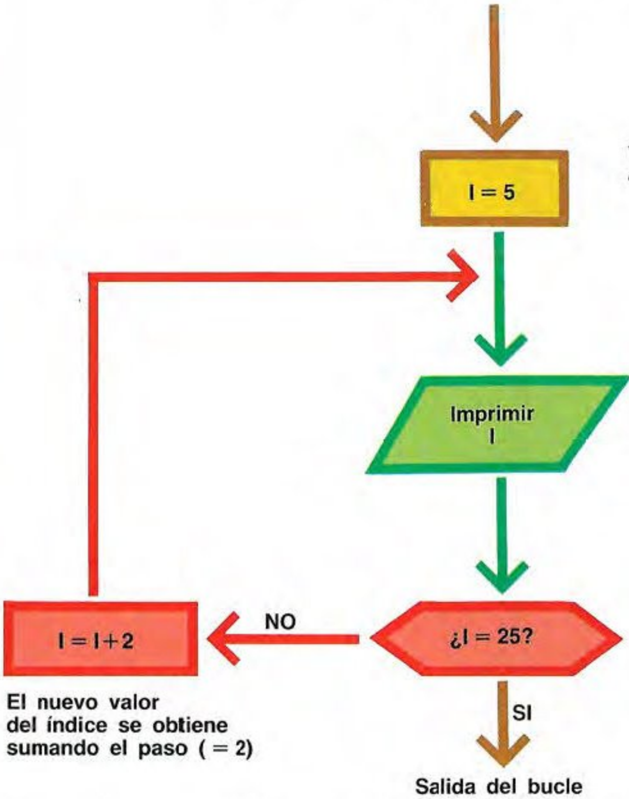
Bucles con paso distinto de 1

El índice de un bucle, es decir, la variable que se incrementa después de cada paso por la parte recurrente (ver gráfico superior de la pág. 187), puede sufrir incrementos distintos de 1. Así, cabe seleccionar sólo algunos de entre los datos disponibles. Por ejemplo, si queremos escribir todos los números impares entre 5 y 25, el valor inicial del bucle es 5, el final 25, y el paso, o sea el incremento del índice, es 2.

En el primer paso, el índice tendrá el valor 5, que es el primer número a imprimir; en el segundo paso el índice será $5 + 2 = 7$, así sucesivamente hasta el valor 25. En el gráfico superior de la página contigua vemos el diagrama de flujo de este bucle. El valor del incremento del índice se denomina normalmente **step** (paso); en el ejemplo anterior tenemos step = 2.

El paso puede ser también negativo; en tal caso, el valor inicial del índice del bucle es mayor que el final, y los valores intermedios se obtienen restando cada vez del índice el valor del paso. En el ejemplo anterior, poniendo: valor inicial 25, valor final 5, paso –2, obtendríamos los mismos números pero en orden decreciente (25, 23, 21... 5); en el gráfico contiguo, abajo, vemos el diagrama de flujo de este tipo de bucle. Como ejemplo de aplicación, supongamos que tenemos 12 nombres y queremos imprimirlos a razón de 4 por línea, en 3 líneas. En el gráfico superior de la pág. 188 se muestra el formato. Los datos a imprimir (nombres del 1 al 12) están contenidos, secuencialmente, en un

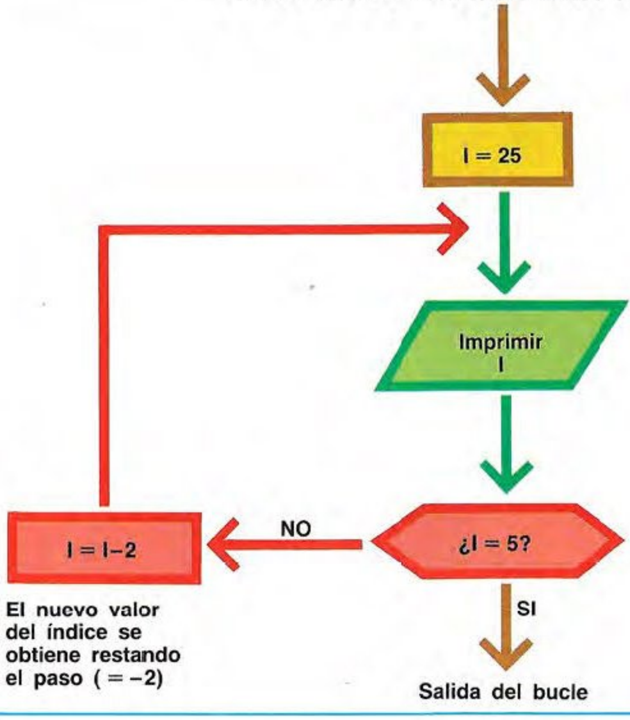
EJEMPLO DE BUCLE CON PASO = 2



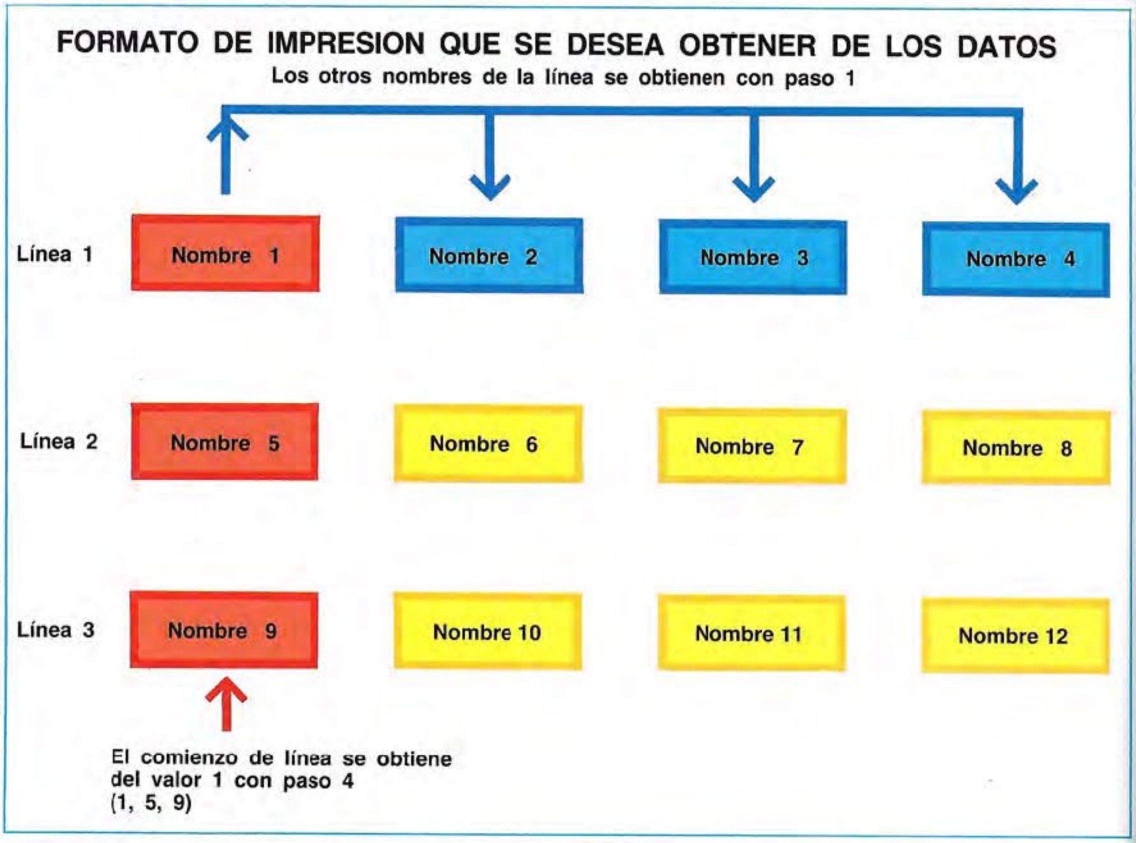
Valor inicial del índice

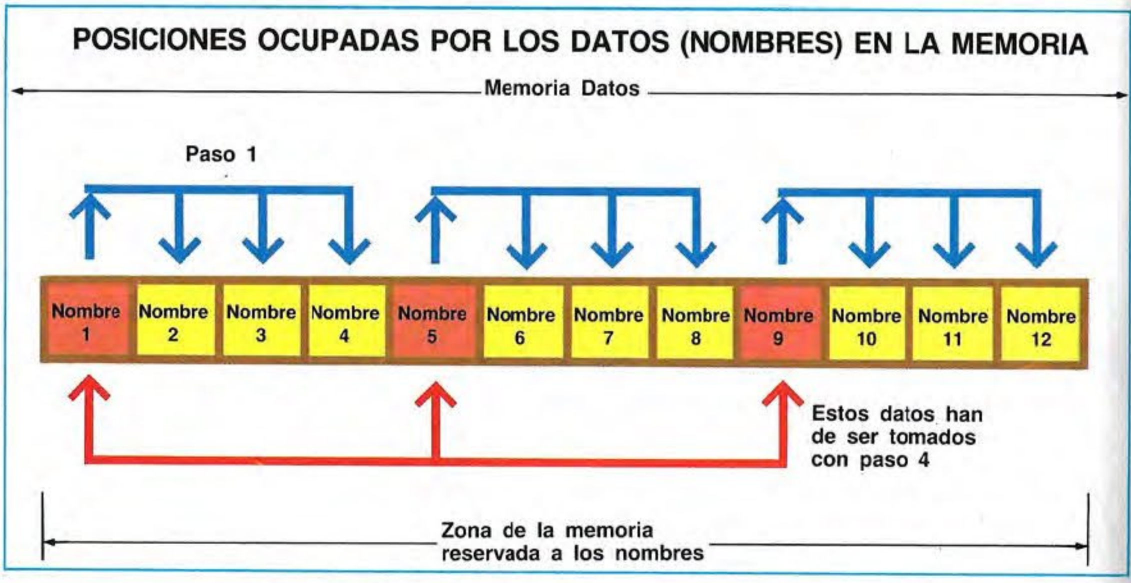
En este ejemplo, (escribir números impares entre 5 y 25), el propio valor del índice es el resultado.
En impresión tendremos la serie de números: 5, 7, 9, 11 ... 25

EJEMPLO DE BUCLE CON PASO = – 2



En este caso, la salida es en orden inverso respecto al anterior: 25, 23, 21 ... 5





área de memoria (ver gráfico). Para obtener la impresión deseada, el procedimiento es:

a / selección del dato de comienzo de línea: nombres 1, 5 y 9, respectivamente, para las líneas 1, 2 y 3

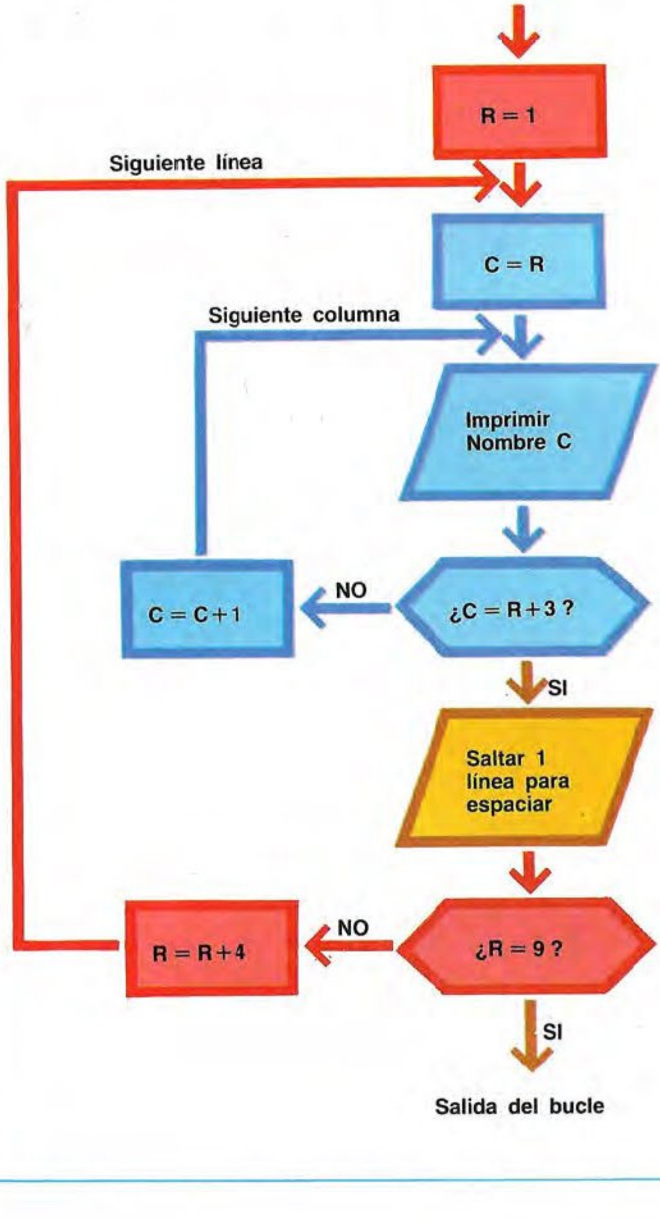
b / impresión del dato seleccionado (nombres 1, 5, 9) y de los tres siguientes sobre la misma línea

c / salto de una línea para espaciar el listado

El punto **a** puede resolverse con un bucle que tenga valor inicial = 1, valor final = 9, paso = 4.

El punto **b** es un segundo bucle que tiene como valor inicial el índice del primero (1, 5, 9) y como final ese mismo índice + 3 (por ejemplo, en la primera fila el inicio es nombre 1 y el final nombre 4). El paso de este bucle es 1.
Para satisfacer el punto **c** es necesario, al terminar la impresión de una línea y antes de la siguiente, enviar la orden de salto de línea (LF, ver códigos ASCII). En el gráfico inferior se muestra el diagrama de flujo de este ejemplo.

DIAGRAMA DE FLUJO PARA LA IMPRESION DE 12 DATOS (NOMBRES) DIVIDIDOS EN 3 LINEAS DE 4 DATOS CADA UNA



Selección de la línea.

En la primera vuelta (R = 1), C tiene el valor inicial 1 y al final 4. Luego se imprimen los nombres 1, 2, 3 y 4.
En la segunda vuelta (R = 5)
C vale: inicio 5, final 8; los nombres impresos son 5, 6, 7 y 8.
En la tercera vuelta (R = 9), C vale: inicio 9, final 12, y los nombres son el 9, 10, 11 y 12.

Este bucle tiene paso 1 (step = 1)

Este bucle tiene paso 4 (step = 4)

Salida forzada de los bucles

Una de las aplicaciones habituales de los bucles es la búsqueda de datos. Supongamos que tenemos memorizados los apellidos y los números de teléfono de nuestros conocidos y que sean en total, por ejemplo, 200 datos (apellido + número). Queremos saber cuál es el número del señor Pérez. Tendremos que escribir un bucle que realice las funciones:

- **a** / lectura de un dato en la memoria
- **b** / comparación entre el apellido Pérez y el que contiene el dato
- **c** / si la comparación es positiva: escritura del número de teléfono
- **d** / si la comparación es negativa: selección del siguiente dato volviendo a partir del punto **a**

El programa puede escribirse utilizando un bucle entre 1 (primer dato) y 200 (último dato contenido en la memoria); el diagrama de flujo correspondiente está en el gráfico de la pág. 191. Este método tiene el defecto de no prever la posibilidad de interrumpir la búsqueda una vez que el dato (número de teléfono del señor Pérez) haya sido encontrado; por lo tanto, se efectúan las operaciones con pérdida de tiempo. Los datos podrían encontrarse en un floppy (diskette flexible) y requerir, por tanto, un tiempo no despreciable para ser leídos (bloque «lectura datos en posición I» en el diagrama de flujo); con el programa estructurado como en el gráfico contiguo, la lectura prosigue hasta que se terminen los datos aunque la operación podría finalizar en el número 3 y la máquina opera durante un tiempo superior al necesario. La reducción de los tiempos a lo estrictamente necesario se logra con un Indicador (**Flag**) que señale el fin de la búsqueda; en este caso las operaciones de lectura y comparación de datos se suspenden y el programa termina mucho antes. En general, el uso de un flag es el siguiente:

- inicialmente el flag se pone a cero, lo que indica que hay que efectuar todas las operaciones previstas
- cuando se obtiene el resultado deseado, el flag toma un valor distinto (por ejemplo, 1), y esto indica que todas las operaciones han de saltarse

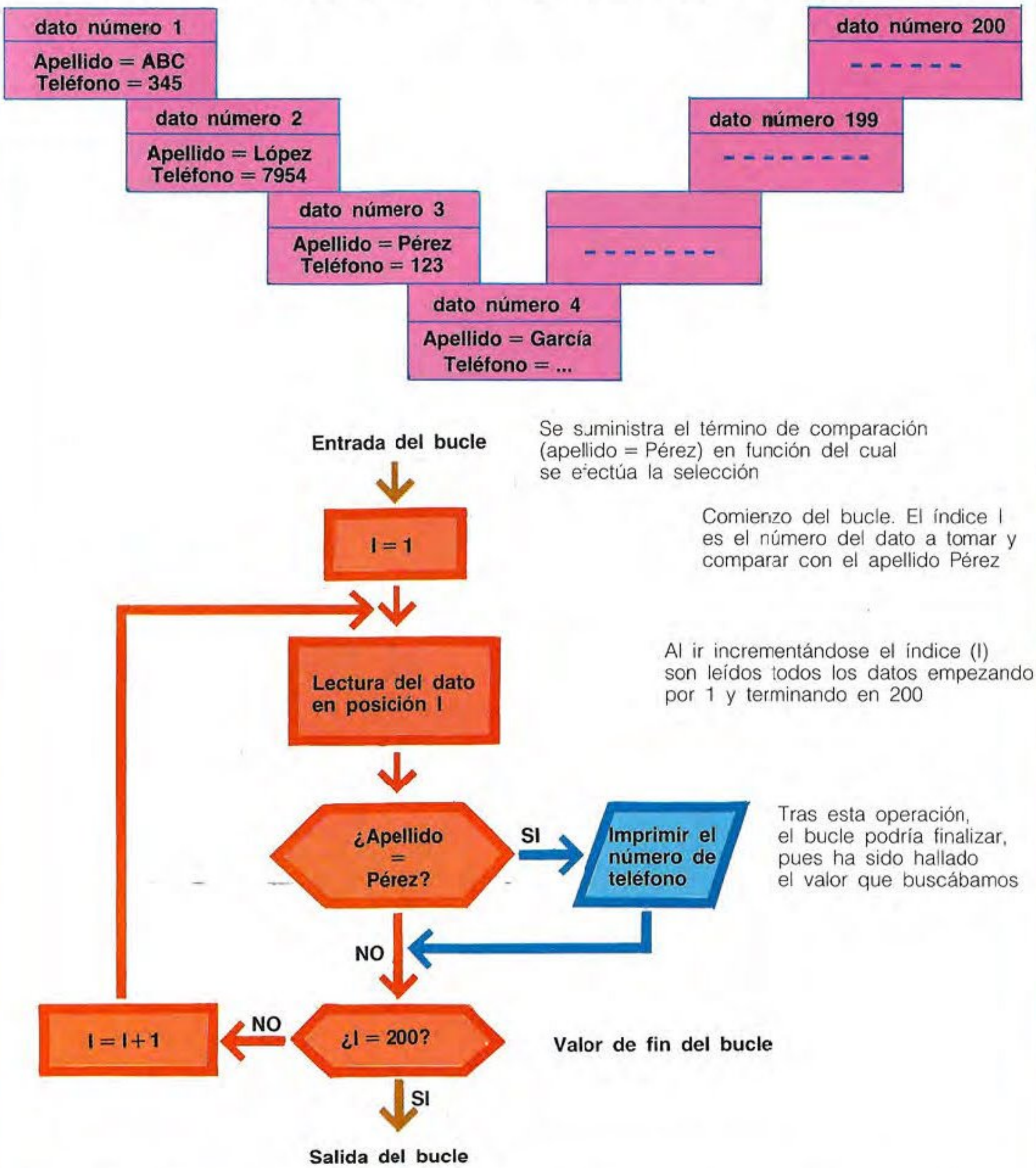
Un operador controla los resultados a la salida de un bucle de búsqueda.



Marka

DIAGRAMA DE FLUJO DE UN BUCLE PARA LA BUSQUEDA DE UN DATO

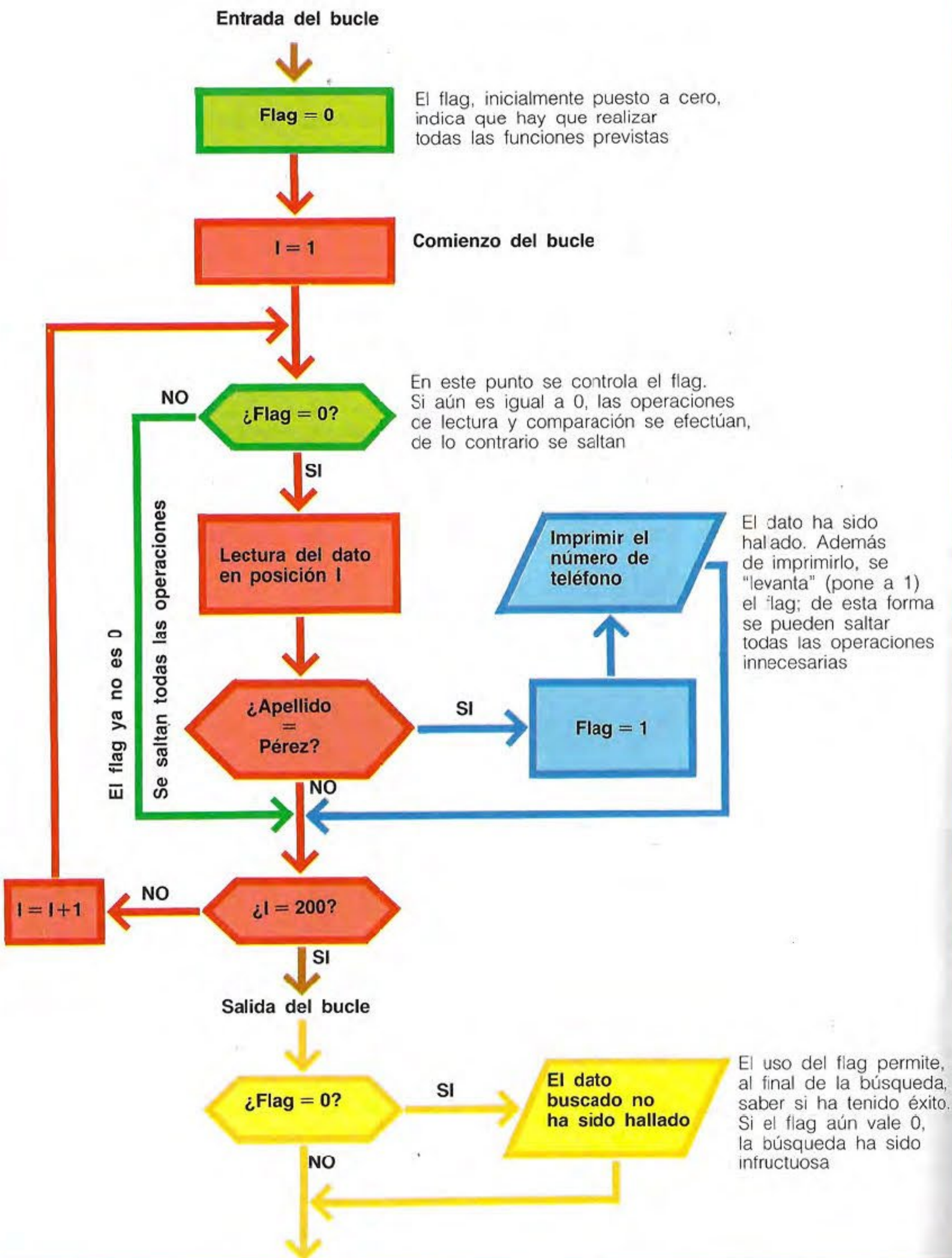
Estructura de datos utilizada en el diagrama. El dato buscado (Pérez, Tel. = 123) es el número 3



En esta forma de bucle no se puede modificar la secuencia de las operaciones, por lo que la búsqueda continúa hasta el final del bucle aunque el dato haya sido hallado.

En el gráfico de la pág. 192 vemos el diagrama de flujo con las modificaciones necesarias para la introducción del flag; su uso, además de ahorrar tiempo, permite conocer el éxito de la búsqueda. En nuestro ejemplo, dicho éxito es inmediatamente visible ya que, en caso positivo, se imprime el número de teléfono; en otros casos, el programa no prevé impresión alguna, luego, si el dato no ha sido hallado, podría haber errores en los cálculos siguientes; el uso del flag permite condicionar las operaciones que siguen al éxito o fracaso de la búsqueda. En otras pala-

DIAGRAMA DE FLUJO DE UN BUCLE DE BUSQUEDA CON FLAG PARA ABREVIAR TIEMPOS ELIMINANDO OPERACIONES INNECESARIAS





E. Ferorelli/Life-Grazia Neri

Con los videojuegos, el ordenador ha entrado en el hogar.

bras, el flag se utiliza como un indicador en función del cual se puede elegir el camino a seguir en el curso del programa.

Aplicaciones de los diagramas de flujo

Los métodos de análisis y la técnica de resolución de problemas mediante diagramas de flujo, además de resolver problemas inherentes al mundo del trabajo, son indispensables también para aplicaciones como la utilización del microordenador como adversario de juego.
También en estas aplicaciones se pueden hallar problemas muy complejos, cuya solución sólo es posible tras un análisis atento y detallado. Utilizando los diagramas de flujo, tras dicho análisis la escritura de la serie de instrucciones que constituirán el programa se convierte en mera cuestión técnica, sin grandes riesgos de error. A título de ejemplo, veamos el procedimiento para desarrollar un programa que permita a la máquina jugar a **Strike and Ball**. Más adelante aprenderemos a traducir estos diagramas en instrucciones; tendremos así a disposición un divertido programa de aplicación.
Antes de analizar el problema desde el punto de vista de la programación, hay que conocer las reglas del juego, que pueden esquematizarse de la forma siguiente:
El contrincante (el ordenador) genera 4 números al azar, que hay que adivinar con un máximo de 10 intentos.
A cada intento, el ordenador ha de responder:
NADA, si no hemos adivinado ningún número;
STRIKE, si hemos adivinado algunos números, pero no su posición; en este caso tiene que decirnos también cuántos números hemos adivinado;
BALL, si hemos adivinado tanto el valor como la posición de algunos números. También en este caso el ordenador ha de decirnos cuántos BALL hemos acertado.
Por ejemplo, la respuesta podría ser 2 STRIKE y 1 BALL, lo que significa que, de los 4 números generados por el ordenador, hemos acertado el valor pero no la posición de 2, y tanto el valor como la posición de 1.

Los circuitos integrados

Los que viven fuera del mundo de la microelectrónica piensan que se trata de un mundo perfecto, en el que ya se ha alcanzado ese límite natural más allá del cual la simple realidad de las cosas se contrapone al ingenio humano. Y no sólo el hombre de la calle lo cree: también los que trabajan diariamente con ordenadores, los especialistas del software y los usuarios más sofisticados creen en esta verdad aparente. Nada más falso.

Hoy por hoy, los circuitos integrados están hechos casi exclusivamente de silicio, pero se utilizan al menos ocho tecnologías básicas diferentes para su fabricación y, todas ellas, además de ventajas, tienen sus inevitables inconvenientes y limitaciones. Nos limitaremos aquí a exponer una breve panorámica de las principales tecnologías hoy utilizadas en la fabricación de chips, intentando mostrar, dentro de lo posible, sus pros y sus contras.

Las ocho tecnologías examinadas se dividen en tres grupos principales: las «bipolares», las «MOS» y las que, en lugar de silicio, utilizan como material básico el arseniuro de galio.

Los semiconductores de mayor difusión son, hoy día, los de silicio, cuya fabricación sigue dos vertientes principales. Por una parte tenemos los componentes «bipolares», en los que la corriente que pasa a través de un componente se separa en dos flujos, cada uno de los cuales se mueve hacia un polo eléctrico. Por otra, tenemos los MOS, contracción de MOSFET (Metal Oxide Semiconductor Field Effect Transistor)*; los MOS son componentes en los cuales un solo flujo de corriente se aprovecha de una forma que permite obtener el efecto transistor.*

Los chips bipolares pueden fabricarse según tres tecnologías diferentes, aunque similares: TTL, ECL e I_2L.

- *Los* TTL (Transistor-Transistor Logic) *son tal vez los semiconductores más antiguos y ampliamente utilizados: junto a su alta velocidad de conmutación* (comprendida entre 1,5 y 3 nanosegundos**), tienen el inconveniente de su gran consumo de energía, lo que significa también una notable disipación de calor. En circuitos complejos, en los que centenares o millares de chips operan simultáneamente, esto puede hacer necesario un sistema especial de refrigeración de las máquinas.*

Dan Silversmith, responsable del desarrollo de los componentes de silicio en el Lincoln Laboratory del MIT, ha definido los TTL como los «caballos de tiro» de la tecnología bipolar, admitiendo que la demanda de estos componentes, que ya tienen 15 años de antigüedad, es aún grande, aunque pocas empresas se sigan dedicando a su fabricación.

La solución TTL tiene la desventaja de una técnica de fabricación complicada. Para hacer un chip TTL hay que fabricar unas 10 o 12 máscaras superpuestas (las máscaras son la reproducción fotográfica de los «caminos» metálicos entre una parte de semiconductores y otra, y de otras características estructurales). La estructura del chip se forma por niveles superpuestos, mediante las máscaras. Cuanto mayor es el número de máscaras, más compleja es la fabricación y, por tanto, más alta la probabilidad de errores o defectos en el producto acabado, con un menor rendimiento y costos más elevados.

- *La tecnología* ECL (Emitter-Coupled Logic) *es pariente próxima de la TTL, pero los chips ECL tienen una velocidad de conmutación notablemente mayor, con valores comprendidos entre 0,4 y 1 nanosegundo. Esta tecnología es, por tanto, idónea para componentes como las memorias, que requieren velocidades de trabajo muy altas. El reverso de la medalla es el consumo, el más elevado de todos los tipos de chips. Otra ventaja de la tecnología ECL es la posibilidad de miniaturización. Efectivamente, es la única tecnología bipolar que puede utilizarse para los chips VLSI* (Very Large Scale Integration) *de la última generación, que tienen una elevadísima densidad de componentes por unidad de superficie. Los TTL y los ECL sólo pueden utilizarse en los chips MSI* (Medium Scale Integration)*, dos peldaños por debajo de los VLSI.*

- I_2L. *La tecnología bipolar más reciente es la I_2L, que ha permitido reducir en parte el problema del consumo.*

Pero al ganar en reducción de consumo y dimensiones, se pierde velocidad: los chips I_2L son los más lentos, con velocidades de conmutación comprendidas entre 2,5 y 10 nanosegundos. Este inconveniente queda parcialmente compensado por su mayor facilidad de fabrica-

---

* La «velocidad de conmutación» determina la velocidad máxima a la que el componente puede trabajar (por ejemplo, la máxima velocidad de ejecución de una operación lógica).
** 1 nanosegundo = 10^{-9} segundos = 0,000000001 segundos.

ción. Los I_2L requieren de 8 a 10 máscaras, menos que los TTL (10 o 12) y los ECL (9 a 11).
La tecnología MOS (Metal Oxide Semiconductor), contracción de MOSFET, es más simple que la bipolar, y requiere un número de máscaras siempre inferior a 10, uniendo a ello la ventaja de un menor consumo de energía. Las tecnologías MOS básicas son tres: NMOS, PMOS y CMOS. En NMOS y PMOS, la N y la P indican el signo de las cargas eléctricas que fluyen en el canal, negativas y positivas respectivamente.
La tecnología CMOS, donde la C significa «comentario», utiliza circuitos «dobles», en los que cada componente está formado por un transistor PMOS y un NMOS. Siendo las cargas de distinto signo, cuando funciona un transistor el otro queda inhibido, y esto hace que, en condiciones estáticas, el consumo sea casi nulo.

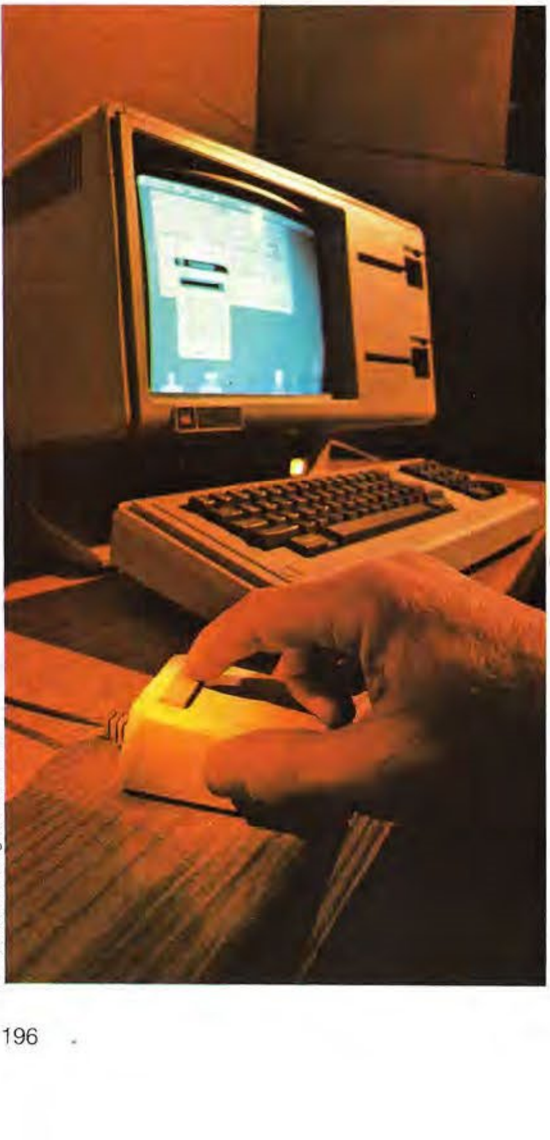
- Los NMOS tienen la ventaja de su reducido tamaño. De una sola oblea de silicio (una lámina de cristal en la que se engloban muchos circuitos integrados iguales y adyacentes) se pueden obtener muchos más circuitos que con otras tecnologías. Esto hace que los NMOS cuesten menos. Y son también los más veloces de la familia MOS, lo que los hace especialmente apetecibles en el mercado de gran distribución. Su desventaja está en el consumo, superior al de los CMOS, puesto que requieren una alimentación estatica mayor. Pero el consumo es, de todos modos, enormemente inferior al de los chips bipolares.
- Los PMOS han sido los primeros circuitos MOS que se fabricaron, y siguen siendo los menos caros, puesto que requieren el menor número de máscaras: de 6 a 8. Su desventaja está en el tamaño, superior al de los NMOS, y en la velocidad. Los PMOS son los chips más lentos de todos, y no compensan esta característica ni siquiera en el consumo.
- Los CMOS son sin duda los más interesantes de la familia, puesto que resultan los más veloces y eficientes. Sin embargo, el primitivo entusiasmo de los fabricantes se ha enfriado un poco debido a los mayores costos de producción. Hay, además, un problema técnico ligado a los CMOS: el del «latch-up»; en determinadas condiciones, los elementos del circuito tienden a bloquearse, y no se puede hacerles cambiar de estado sino dejando el circuito sin alimentación. Para obviar este inconveniente hay que aumentar el aislamiento entre el transistor PMOS y el NMOS de cada elemento del circuito, pero ello implica costos de producción aún mayores. Y los costos son la única explicación de la escasa difusión que han tenido esta clase de componentes.

La tecnología que se basa en el silicio como soporte y componente fundamental, no es la única alternativa en el campo de los semiconductores. Existen otras posibilidades, y el arseniuro de galio en sustitución del silicio es, sin duda, «el camino del futuro».
Este compuesto, cuya fórmula es GaAs, permite producir transistores muy veloces, pero hoy por hoy tiene todavía el problema del precio: es extremadamente caro. Una lámina de GaAs cuesta cien veces más que una oblea de silicio. Y además es difícil conseguir la formación de cristales de GaAs cilíndricos, que es la forma normal de los cristales de silicio y la más adecuada para la elaboración.
Otro problema potencial está ligado al compuesto mismo. El GaAs está formado por la combinación de dos elementos, galio y arsénico. El arsénico es muy tóxico, y podrían plantearse problemas nada desdeñables al trabajar con grandes cantidades, problemas inexistentes en el caso del silicio.
Una alternativa que permite alcanzar el objetivo de un bajo consumo y una velocidad elevada la ofrece, no una nueva tecnología, como en el caso del GaAs, sino el operar en distintas condiciones ambientales. Sumergiendo los circuitos en un baño de helio líquido, se puede llevar su temperatura cerca del cero absoluto, y en estas condiciones «ambientales» se manifiesta el fenómeno de la «superconductividad». Los diversos elementos del circuito tienden a perder casi por completo toda resistencia pasiva y, consiguientemente, se vuelven muy veloces.
Los circuitos capaces de operar en estas condiciones se denominan de «unión Josephson», en honor del investigador que descubrió la superconductividad en 1962. Un circuito de unión Josephson puede conmutar en unos 100 picosegundos, con una velocidad, pues, de 10 a 100 veces superior a la de los circuitos normales.
IBM ha realizado ya prototipos de procesadores Josephson, pero, según Silversmith, pasarán por lo menos diez años antes de que estos productos puedan estar en el mercado. Y en diez años pueden suceder otras muchas cosas.

(Condensado de «La jungla de las siglas», en la revista italiana COMPUTER, nº 53, octubre de 1982)

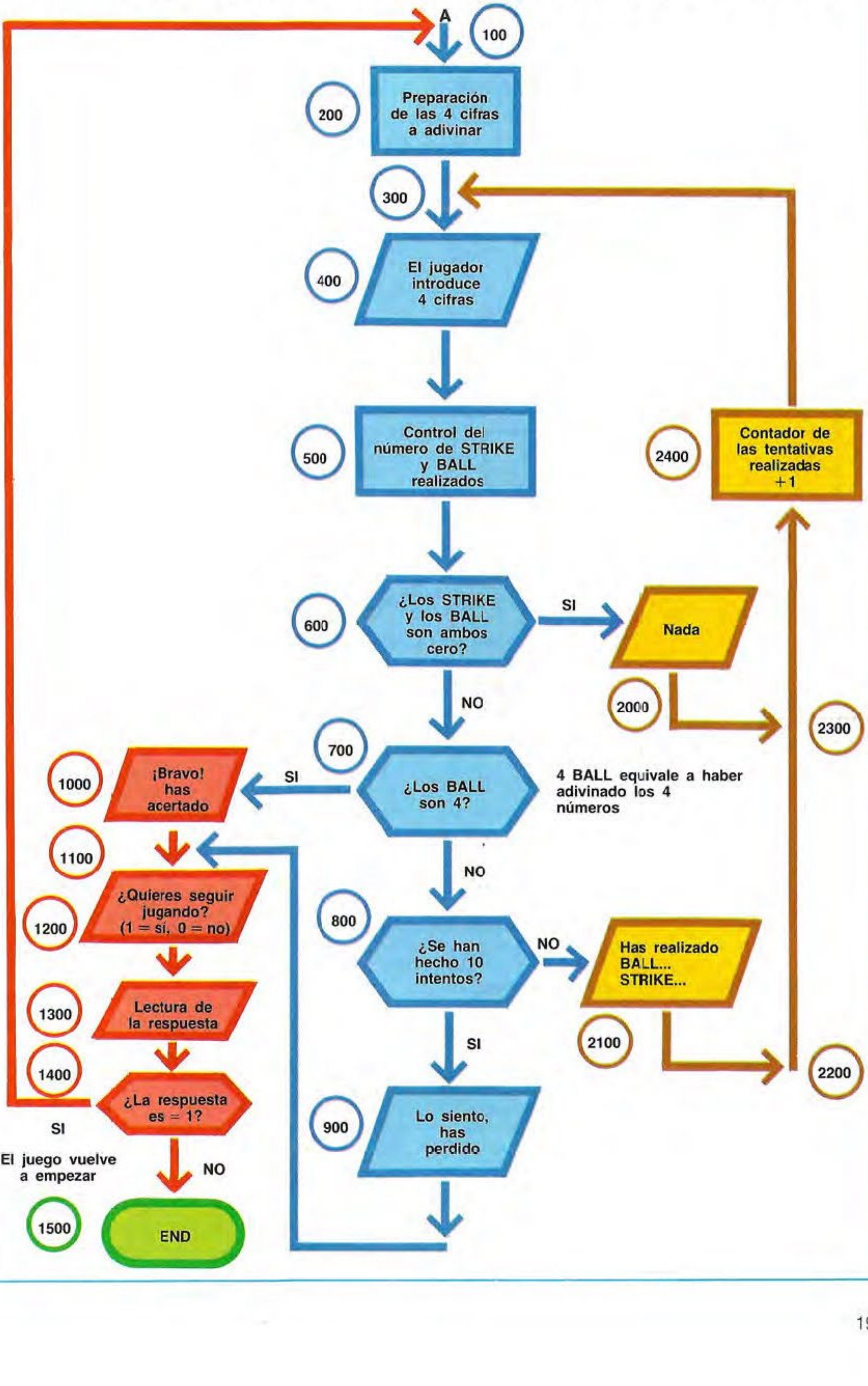
La exposición de las reglas del juego constituye, de hecho, el análisis del problema.
Este sencillo método, que consiste en la **exposición clara, detallada y secuencial de las funciones a realizar**, permite analizar y resolver cualquier clase de problema, y es la base de preparación de los diagramas de flujo.
Traduciendo la anterior esquematización descriptiva a la simbología de los diagramas de flujo, habremos terminado el análisis de primer nivel. En el gráfico de la página contigua se muestra el correspondiente diagrama de flujo.
En los diagramas es conveniente, al principio, poner «etiquetas» numéricas junto a cada símbolo ya que facilita la lectura del diagrama y acelera la fase de escritura del programa.
En el gráfico vemos tres líneas principales: la primera, identificada por números inferiores a 1.000, es el flujo principal; la segunda, de numeración comprendida entre 1.000 y 1.500, es el flujo que termina la partida en curso y comienza otra; la última línea, de numeración entre 2.000 y 2.400, es el flujo de las respuestas que deberá dar la mácuina.
Por ejemplo, el bloque 600 es una decisión; en él la máquina controla si se verifica la situación: «número de STRIKE = 0 y número de BALL = 0». Si esta situación es cierta, el jugador no ha obtenido ningún punto.
Obsérvese que la frase: «número de STRIKE = 0 y número de BALL = 0» equivale a: «STRIKE = 0 AND BALL = 0», en donde el operador lógico AND se usa para controlar la verificación simultánea de dos situaciones.
Si escribiéramos el programa de la forma presentada en el diagrama de flujo, tendríamos todas las funciones necesarias, pero nuestro programa sería «esquelético», poco divertido y, si el jugador no conociera las reglas, inutilizable.
Antes de entrar en el cuerpo principal, que comienza en el punto A y termina en END (bloque 1.500), es necesaria una parte descriptiva que el ordenador pueda presentar a petición del jugador para informarlo sobre las reglas que rigen el juego.
Esta parte inicial de información sobre los objetivos del programa y la forma de utilizarlo es indispensable en todas las aplicaciones.
Normalmente, el usuario de un programa (o el programador mismo) no puede recordar cuáles son los parámetros que hay que suministrar y las posibilidades o funciones de un determinado programa. La parte descriptiva inicial cumple la función de recordatorio y facilita la tarea de quien utiliza el programa.
En el gráfico de la pág. 197 vemos también dos bloques (200 y 500) que realizan funciones complejas, pero no se indica cómo se llevan a cabo dichas funciones. Este nivel de detalle se desarrollará en diagramas sucesivos, dedicados a tal fin. Posponiendo el desarrollo de los detalles a una fase sucesiva se obtienen dos resultados importantes: no se complica demasiado el diagrama principal, facilitando su legibilidad, y, sobre todo, las funciones más complejas se desarrollan por separado. Eso permite escribirlas de forma generalizada, para su eventual uso en otras aplicaciones.
Esta técnica de fraccionamiento de los programas y de escritura de las partes en forma generalizada, es la regla de oro de la programación.

Sugestivo encuadre de un ordenador con el monitor en funcionamiento.



C. O'Rear/West Light-Grazia Neri

DIAGRAMA DE FLUJO DE PRIMER NIVEL PARA EL JUEGO STRIKE AND BALL



En el gráfico inferior se muestra el diagrama de flujo del módulo que cuenta los puntos obtenidos dintinguiendo entre STRIKE y BALL.
Todas las numeraciones de este bloque han de estar comprendidas entre 500 (valor asignado en el diagrama general) y 590 (pues el valor 600 forma parte ya de otro bloque).
Para completar los diagramas del programa queda sólo por desarrollar el módulo identificado por el número 200 (ver gráfico contiguo), que tiene por objeto la preparación de los cuatro números a adivinar.

DIAGRAMA DE FLUJO DETALLADO DEL BLOQUE 500

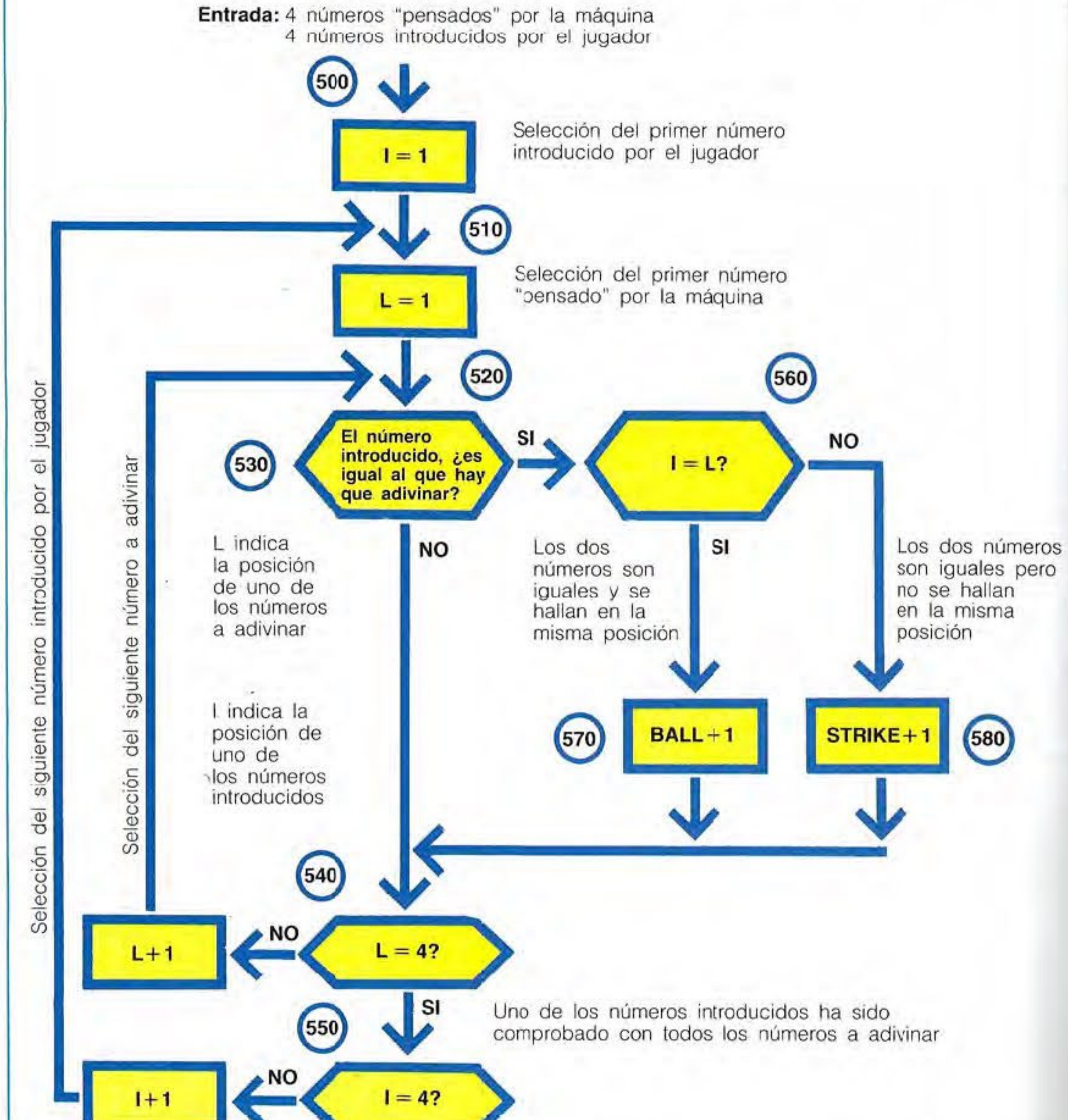
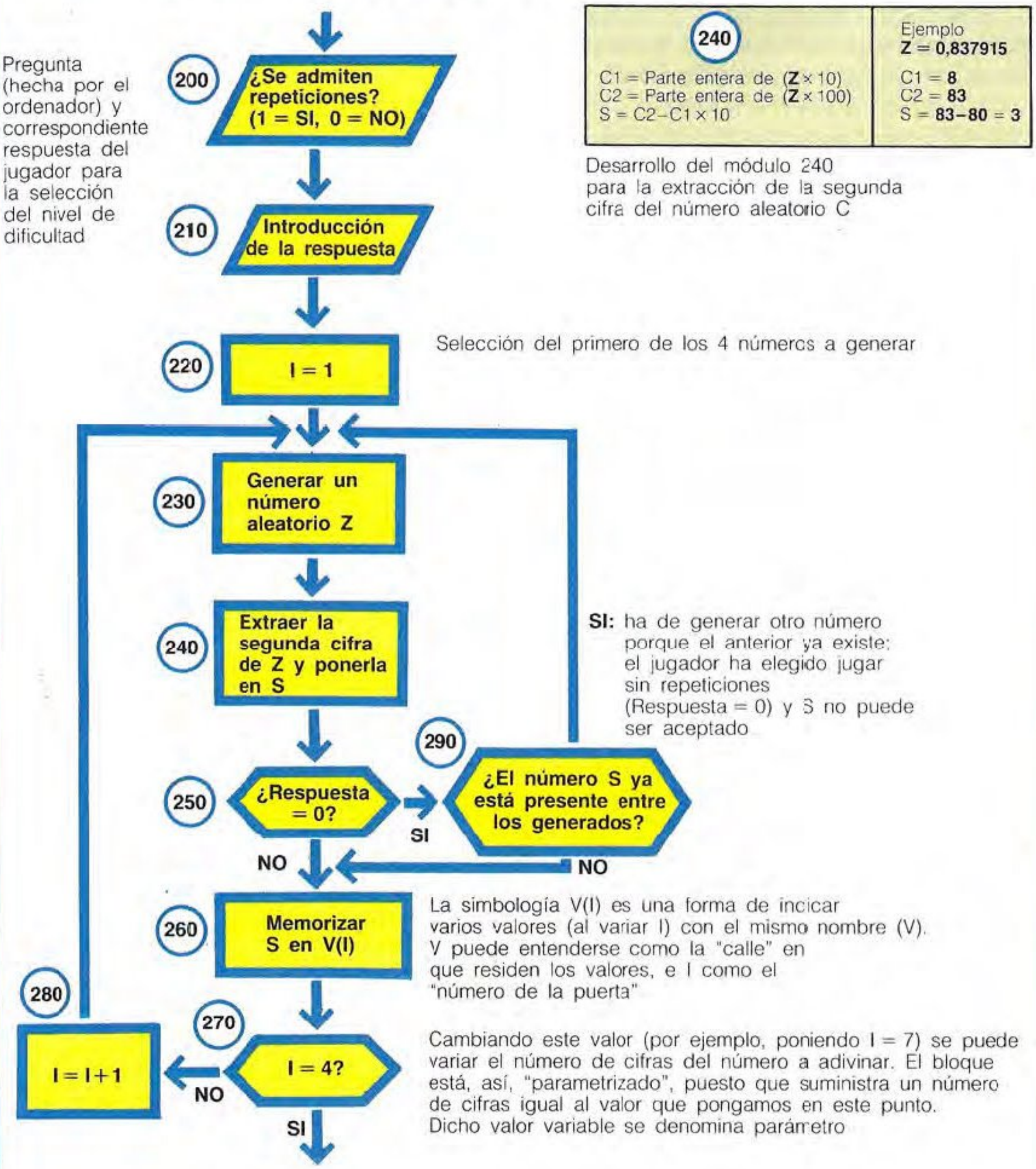


DIAGRAMA DE FLUJO DETALLADO DEL BLOQUE 200



En la salida tendremos:
V(1) = primer número, obtenido en la primera vuelta (I = 1)
V(2) = segundo número (I = 2)
V(3) = tercer número (I = 3)
V(4) = cuarto número (I = 4)

La función principal de este módulo es la de generar números aleatorios, es decir, que no guarden ninguna relación entre sí. De momento nos limitaremos a señalar que en los ordenadores existe una instrucción especial capaz de generar un número aleatorio comprendido entre 0 y 1. Qué es realmente un número obtenido de esta manera y en qué medida se aparta de un verdadero número aleatorio, se verá más adelante. En esta aplicación el resultado final es el de obtener un número de cuatro cifras, es decir, el número a adivinar. Dicho resultado se podría

obtener, simplemente, aislando las cuatro primeras cifras del número aleatorio generado. En este ejemplo, sin embargo, se ha escogido un método más largo, que consiste en generar cuatro números, aislando de cada uno de ellos una de las cifras que hay que adivinar.

En realidad, esta elección lleva a un programa sólo en apariencia más largo, puesto que, con este método, el control de la existencia de eventuales cifras duplicadas en el número a adivinar es más sencillo. Supongamos, por el momento, que la máquina pueda suministrarnos un número aleatorio de seis cifras comprendido entre 0 y 1. Puesto que sólo nos interesa una de estas cifras, tendremos que multiplicar el número aleatorio por 10 y tomar la parte entera.

Por ejemplo, sea 0,837915 el número aleatorio generado; multiplicándolo por 10 tenemos 8,37915, y tomando únicamente la parte entera queda 8. Análogamente, si quisiéramos tomar la segunda cifra (3) en lugar de la primera, tendríamos que:

1 / Extraer la primera cifra como en el caso anterior (obteniendo la cifra 8)

2 / multiplicar el número aleatorio por 100 ($0{,}837915 \times 100 = 83{,}7915$) y tomar la parte entera (83)

3 / multiplicar por 10 el número obtenido en el paso 1 ($8 \times 10 = 80$)

4 / calcular la diferencia entre el número obtenido en el paso 2 y el hallado en el paso 3 ($83–80 = 3$); el resultado es la segunda cifra del número aleatorio inicial

La cifra así obtenida a partir del número aleatorio es una de las cuatro que componen el número que hay que adivinar; repitiendo el proceso cuatro veces y memorizando los resultados, obtendremos el número de cuatro cifras. En este momento tendríamos que decidir qué grado de dificultad queremos dar al juego: en este caso, si se admiten repeticiones entre los cuatro números a adivinar. La pregunta se planteará al comienzo del juego, dejando la decisión al usuario. En el gráfico de la pág. 199 se muestra el diagrama de flujo del módulo 200 desarrollado para componer el número a adivinar tomando la segunda cifra de una serie de números aleatorios, con la posibilidad de excluir repeticiones dentro del número a adivinar.

Para completar la tarea bastará con traducir los diagramas de flujo a cualquier lenguaje simbólico. La lista de las instrucciones de este programa se muestra más adelante.

Aplicaciones científicas

Los diagramas de flujo se aplican, aunque en menor medida, a la resolución de problemas científicos. En estas aplicaciones, normalmente, los programas son estructuralmente sencillos, es decir, compuestos por un número de pasos relativamente bajo; sin embargo, las fórmulas a utilizar pueden ser complicadas.

La estructuración minuciosa del programa pierde importancia, mientras que se vuelve decisivo el análisis de los procedimientos matemáticos y de las secuencias de cálculo. El resultado del análisis no es ya un diagrama de flujo, sino el método matemático de cálculo (algoritmo) que hay que adoptar para resolver un problema específico. En los programas de gestión, sin embargo, el algoritmo de cálculo se conoce siempre con exactitud.

En los problemas científicos, la dificultad mayor estriba precisamente en determinar qué fórmula hay que aplicar y de qué manera. El desarrollo de los cálculos puede producir errores según la secuencia en la que se efectúen los cálculos mismos. Esto se debe a que, inevitablemente, el ordenador ha de redondear cada operación. En los circuitos destinados al cálculo pueden manejarse números de una determinada longitud y no más, y la parte excedente resulta ignorada, lo cual puede producir errores.

Por ejemplo, supongamos que tenemos un ordenador que puede manejar números de un máximo de cinco cifras y queremos efectuar el cálculo: $100 \times 0{,}0002/100$; el resultado debería ser 0,0002, puesto que multiplicando y luego dividiendo un número por 100, permanece invariable.

Realicemos el cálculo efectuando primero la multiplicación:

$$\mathbf{100 \times 0{,}0002 = 0{,}02}$$

El resultado puede ser manejado por la máquina, puesto que tiene menos de cinco cifras. Dividiendo ahora por 100:

$$\mathbf{0{,}02/100 = 0{,}0002}$$

tenemos un número de cinco cifras, y el resultado es exacto. Realicemos ahora el mismo cálculo empezando por la división:

$$\mathbf{0{,}0002/100 = 0{,}000002}$$

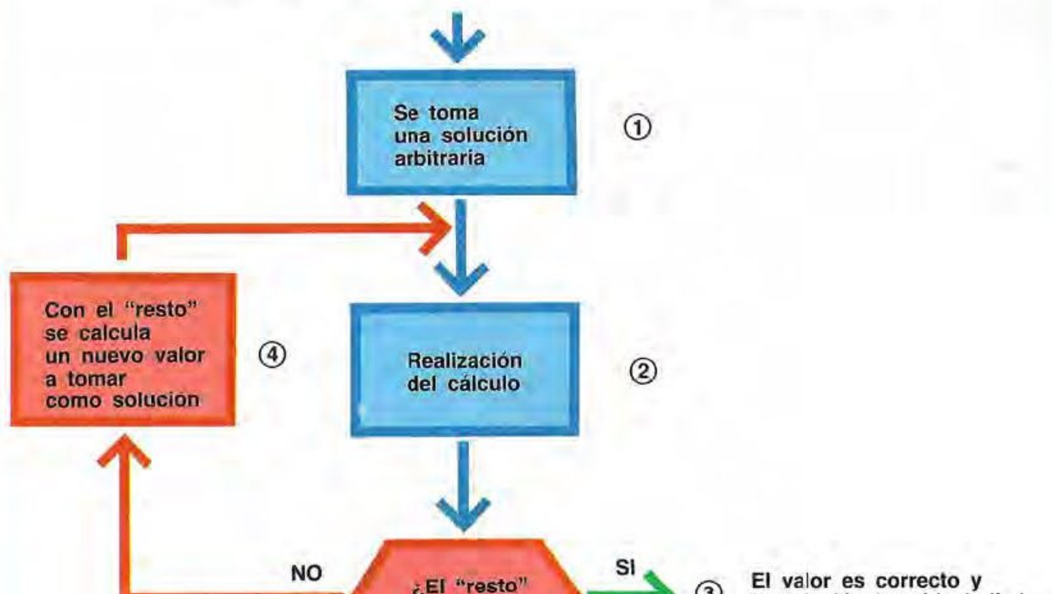
El número 0,000002 tiene más de cinco cifras, por lo que las cifras de la sexta en adelante son automáticamente ignoradas por el ordenador. El resultado de la división se queda en 0,0000 y al final todo el cálculo será erróneo.
Naturalmente, no hay máquinas con un número tan limitado de cifras útiles; la precisión estándar es siempre mucho mayor; pero, de todos modos, en algunas aplicaciones científicas hay que considerar en el análisis la eventualidad de errores debidos al redondeo. Concretamente, si algunos cálculos se repiten gran cantidad de veces, se puede tener una gradual acumulación de errores, que puede dar lugar a inexactitudes en los resultados, incluso en las grandes máquinas de gran precisión. Esta situación es la que se presenta en la resolución de problemas con el método iterativo.
El método iterativo permite hallar la solución del problema formulando inicialmente un valor al azar de la incógnita y modificando dicho valor en función del resultado del propio cálculo.
El esquema básico se muestra en el gráfico inferior. En el paso **1** se escoge un valor al azar, por ejemplo 0; el paso **2** consiste en sustituir este valor en la fórmula resolutiva y verificar (paso **3**) si el desarrollo del cálculo da el valor exacto indicado por la fórmula. En este caso, el valor hipotético atribuido a la incógnita es correcto. En caso contrario hay un «resto» que, utilizado para suministrar un nuevo valor de intento, es, a su vez, introducido en el módulo **2** para una nueva comprobación. El proceso continúa, por aproximaciones sucesivas, hasta obtener un «resto» igual a cero (o muy próximo a cero).
A título de ejemplo, apliquemos el método iterativo a un problema de pesadas.
Un recipiente de peso desconocido contiene 10 bolas, también de peso desconocido. Puesto el recipiente sobre el plato de una balanza, ésta señala un peso total (recipiente + 10 bolas) de 24 kg. Quitamos 5 bolas y obtenemos un nuevo peso de 14 Kg (recipiente + 5 bolas). Hallar cuánto pesa el recipiente y cada una de las bolas.
Llamemos x al peso del recipiente e y al de una bola.
En la primera pesada tenemos:

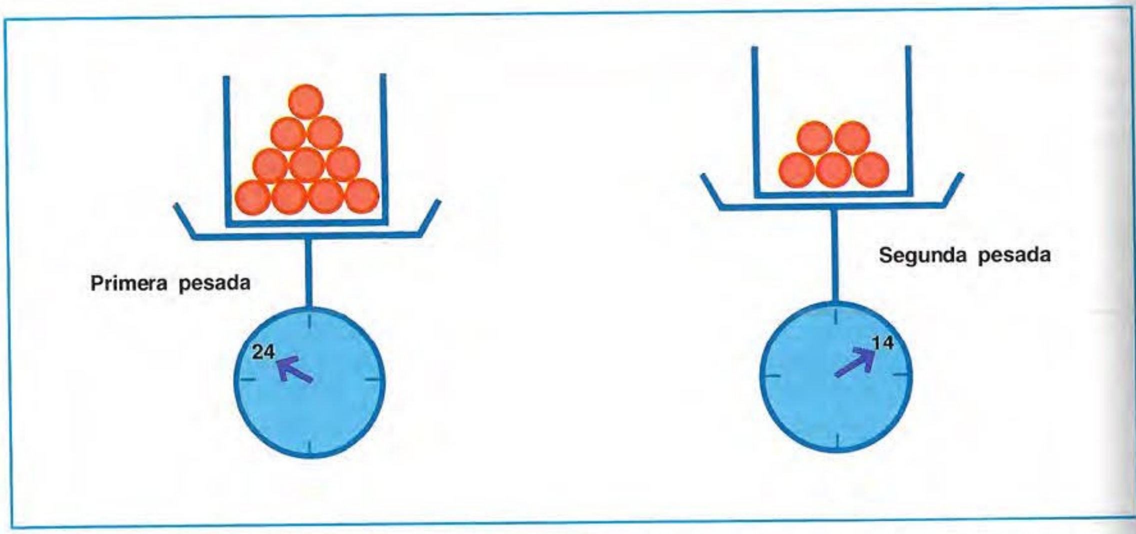
recipiente + 10 bolas = 24 Kg
es decir: $x + 10y = 24$

En la segunda pesada:

recipiente + 5 bolas = 14 Kg
es decir: $x + 5y = 14$

ESQUEMA DE PRINCIPIO DEL METODO ITERATIVO





La solución del problema consiste en resolver el sistema algebraico formado por las dos ecuaciones anteriores:

$$x + 10y = 24$$
$$x + 5y = 14$$

En este caso, el problema se puede resolver con métodos algebraicos normales (por ejemplo, restando las dos ecuaciones), que dan $x = 4$ e $y = 2$. Verifiquemos ahora estas soluciones con el método iterativo:

1 / Asignamos a una de las incógnitas un valor arbitrario, por ejemplo, $x = 0$

En caso de error en los programas de iteración, se puede quedar literalmente "sumergido" en un mar de papel.



M. Wolf/Black Star - Grazia Neri

2 / Este valor se sustituye en la primera ecuación, lo que da:

$$0 + 10y = 24$$

de donde:

$$10y = 24$$

$$y = \frac{24}{10} = 2{,}4$$

3 / El valor así calculado ($y = 2{,}4$) se sustituye en la segunda ecuación, de la que se puede sacar un nuevo valor de x. La ecuación $x + 5y = 14$, con $y = 2{,}4$ da:

$$x + 5.2{,}4 = 14$$

es decir:

$$x + 12 = 14$$
$$x = 14 - 12 = 2$$

4 / Este nuevo valor de x se sustituye en la primera ecuación (como en el punto 2):

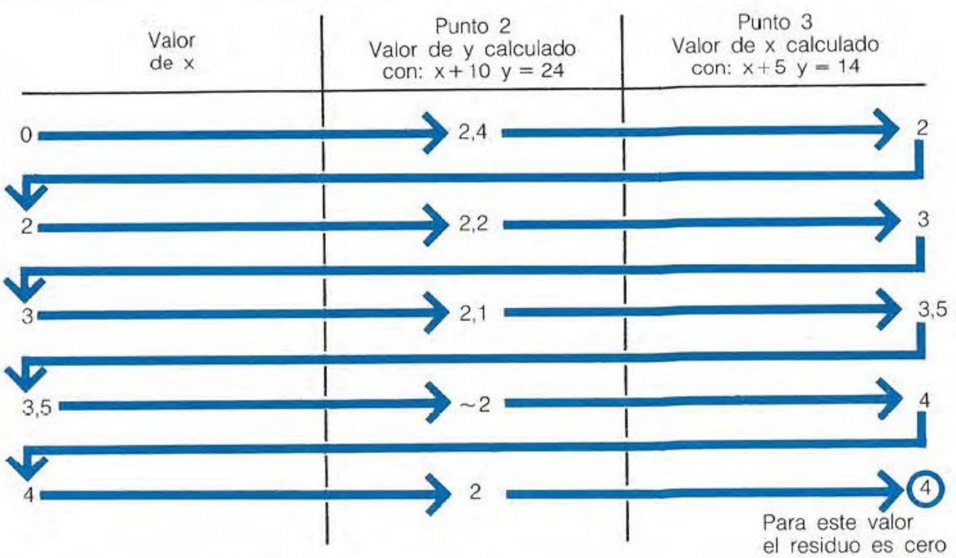
$$2 + 10y = 24$$
$$10y = 24 - 2$$
$$10y = 22$$

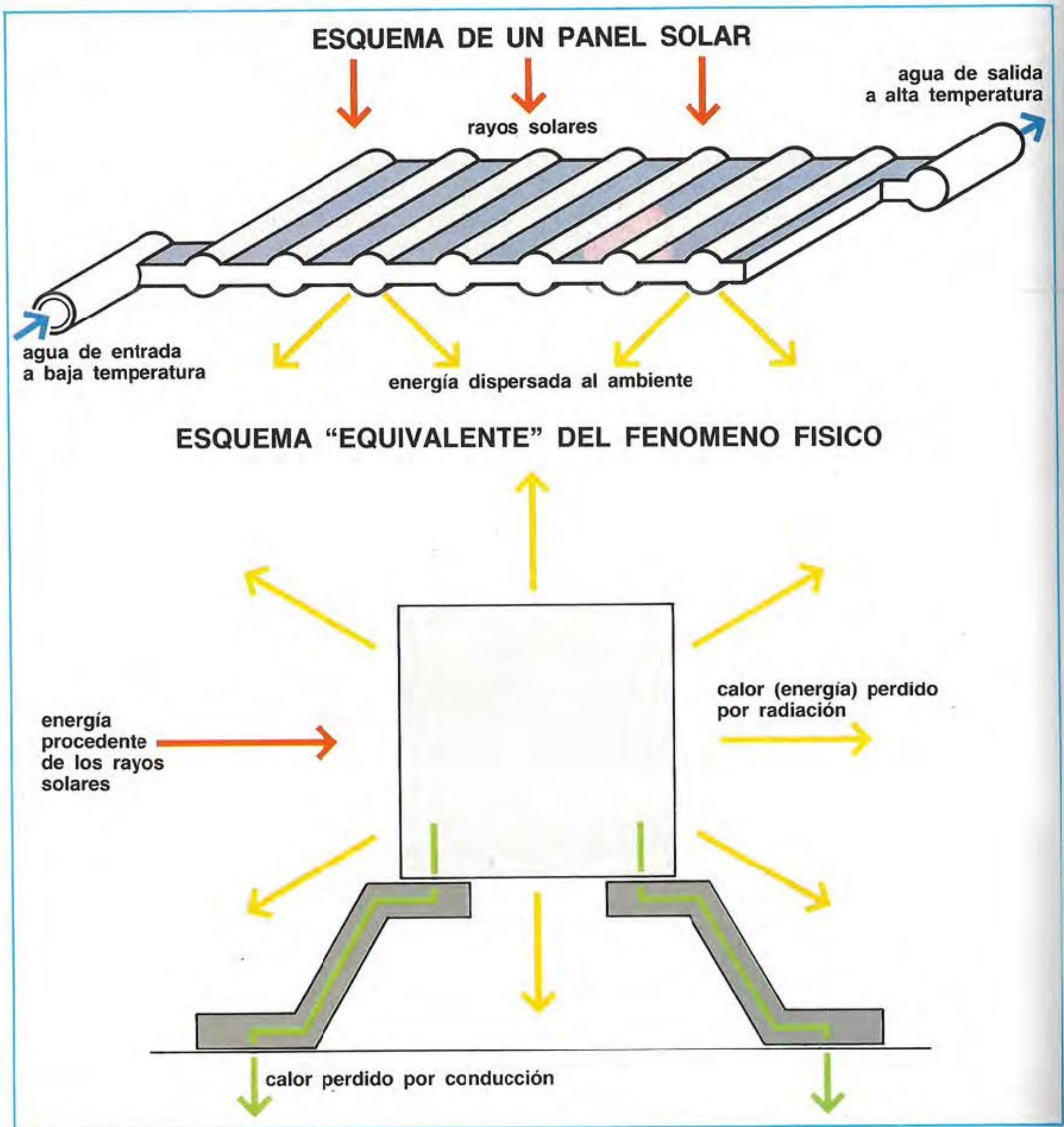
$$y = \frac{22}{10} = 2{,}2$$

Pasos sucesivos como en los puntos 2, 3 y 4, dan los valores del gráfico inferior.
Las dos ecuaciones $x + 10y = 24$, $x + 5y = 14$ dan un resto nulo poniendo $x = 4$ e $y = 2$.
Por tanto, estos dos valores son la solución buscada. Si se invirtieron los términos del problema, es decir, primera pesada con 5 bolas y segunda con 10, el método, en lugar de «converger» dando soluciones cada vez más aproximadas, daría valores negativos «divergentes», es decir, cada vez más alejados de la solución. Cuando se adoptan estos métodos hay que analizar muy atentamente el planteamiento del cálculo para evitar estructuras que llevarían a la divergencia de las soluciones por tanteo.
El ejemplo propuesto es banal, puesto que el sistema posee soluciones algebraicas y, por tanto, no hay motivo para utilizar métodos iterativos para buscar las soluciones. En las aplicaciones científicas, sin embargo, se presentan a menudo problemas que no pueden resolverse sino con métodos de cálculo similares al que acabamos de ver. Dada la importancia del tema, se han desarrollado numerosas teorías, algunas notablemente complejas, sobre los métodos a adoptar para garantizar la convergencia. Acto seguido veremos un ejemplo de aplicación del método iterativo para la resolución de problemas térmicos.

Aplicaciones del método iterativo

El ejemplo anterior (resolución de un sistema de dos ecuaciones con dos incógnitas) ha sido expuesto sólo a título explicativo; en ningún caso conviene utilizar métodos iterativos cuando hay solución albegraica. Por más complejos que sean, los sistemas de primer grado (en los que





las incógnitas no están elevadas a potencia alguna) con un número de ecuaciones igual al número de las incógnitas, pueden resolverse con métodos algebraicos, más rápidos y seguros que los métodos iterativos.

Hay, sin embargo, algunos tipos de ecuaciones que no tienen soluciones algebraicas, y el único camino a seguir con ellos es el método iterativo. Una ecuación típica de esta clase es:

$$A \cdot T^4 + B \cdot T = C$$

La incógnita es T, mientras que A, B y C son valores numéricos constantes. Antes de examinar uno de los métodos de resolución de esta ecuación, veamos en qué campos se aplica.

Conocemos los acuciantes problemas energéticos de la civilización industrial. Los recursos tradicionales, sobre todo el petróleo, están agotándose, su costo aumenta cada vez más y producen contaminación. Por estos motivos es urgente hallar fuentes de energía alternativas. Una de éstas, tal vez la principal, es la energía solar. Todos sabemos que el agua que contiene un recipiente expuesto a los rayos del sol, se calienta; este efecto es consecuencia de la acu-

mulación de la energía recibida de los rayos solares. Una vez calentada el agua, podemos aprovechar esta energía de muchas maneras. La manera más simple consiste, obviamente, en usar el agua para las necesidades domésticas. La aplicación de las fuentes de energía alternativas es ya una realidad, y hay en el comercio «paneles solares» que realizan las funciones citadas. Para obtener un buen rendimiento de la instalación hay que proyectarla cuidadosamente, para lo que se utiliza la fórmula expuesta.

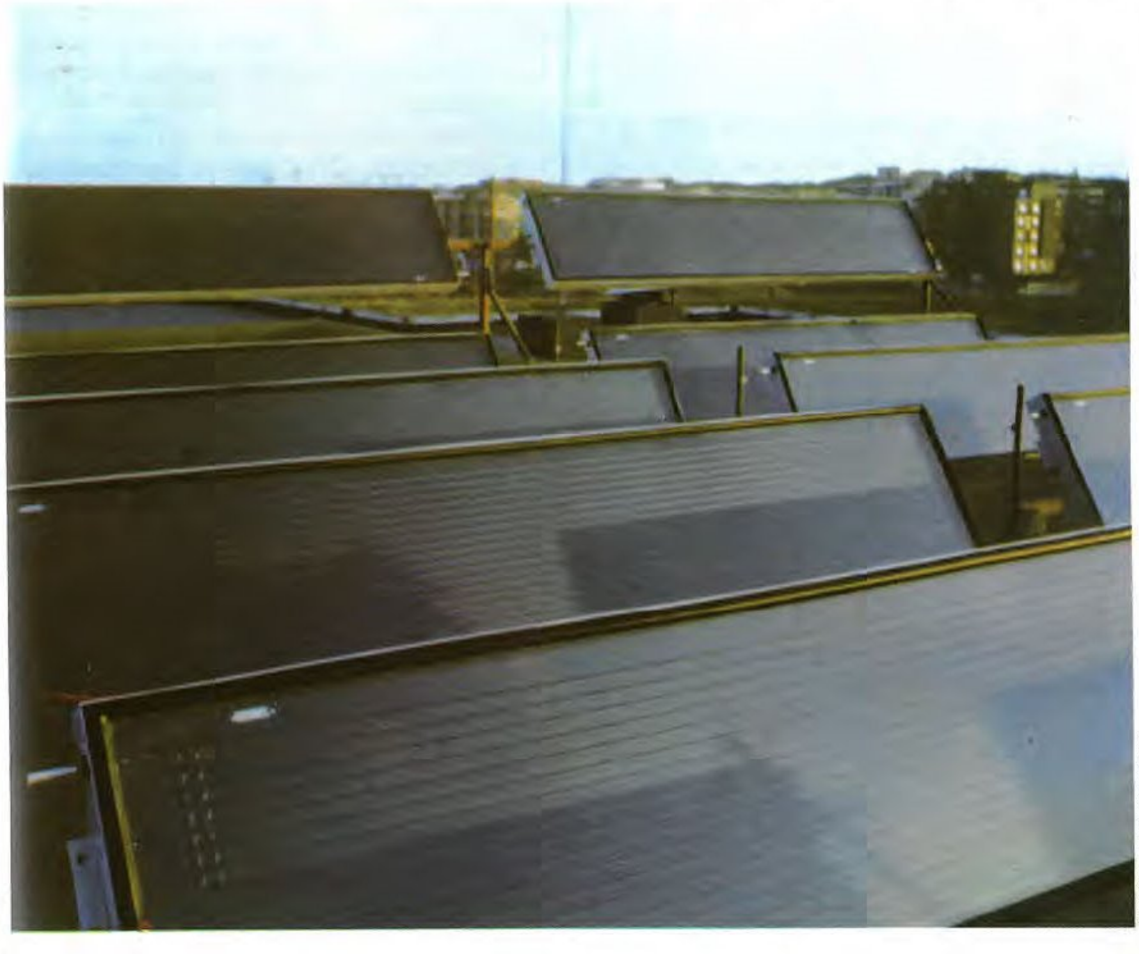
El fenómeno físico se puede esquematizar, en una primera aproximación, como se ve en el gráfico de la pág. 204. Arriba vemos el esquema funcional de un panel solar; abajo, una esquematización que, en determinadas condiciones, equivale al fenómeno físico real. Supongamos que el agua no circule. Si quisiéramos mantener el agua contenida en el recipiente (sin circular) a la temperatura T, nos limitaríamos a suministrar tanta energía como pierde el depósito: si suministramos menos, el agua se enfría; si suministramos más, se calienta. En condiciones «estacionarias» de agua quieta y temperatura constante, el balance energético es:

energía suministrada = energía perdida

La energía suministrada es la cantidad de calor cedida por los rayos solares, y llamamos C a su valor. La energía perdida es la que el recipiente cede al ambiente circundante (pérdidas). Las principales pérdidas térmicas son las que se producen por conducción y por radiación. Las pérdidas por conducción se deben a la cantidad de calor que pasa de un cuerpo más caliente a uno más frío en contacto con él, y son proporcionales a la diferencia de temperatura entre los dos cuerpos:

pérdidas por conducción =
= B·(temperatura del primer cuerpo−
− temperatura del segundo cuerpo)

Estas superficies, que a primera vista pueden parecer pizarras, son en realidad paneles solares.



Archivio Curcio/CTIP Solar

Para simplificar (cuanto estamos viendo tiene sólo un carácter indicativo), despreciemos la temperatura del segundo cuerpo con respecto a la del primero:

pérdidas por conducción = B·T

B es un factor que depende del tipo de material y de la forma geométrica del sistema.

El segundo tipo de pérdidas, las que se producen por radiación, se debe a la fuga de calor de un cuerpo hacia el ambiente circundante (más frío) bajo la forma de ondas electromagnéticas (radiaciones, de ahí el término «radiación»). En este tipo de intercambio de calor, la energía transmitida es proporcional a las diferencias de las cuartas potencias de las temperaturas:

pérdidas por radiación =
= A·[(temperatura del primer cuerpo)^4–
– (temperatura segundo cuerpo)^4]

Despreciando también en este caso la temperatura del segundo cuerpo, tenemos:

pérdidas por radiación = $A·T^4$

El factor A depende de las características físicas y geométricas del cuerpo.

La energía total perdida será la suma de las dos pérdidas (conducción + radiación):

energía total perdida = $A·T^4 + B·T$

Para mantener constante la temperatura (T) hay que suministrar al depósito una cantidad igual de energía; en otras palabras, la energía aportada por los rayos solares (C) ha de ser igual a la energía total perdida:

$$C = A·T^4 + B·T$$

La ecuación obtenida es una forma simplificada, puesto que no tiene en cuenta las pérdidas por intercambio de calor con el aire en contacto con el depósito (intercambio por «convección»); además, en muchos casos prácticos, si el valor de T no es grande, el termino $A·T^4$ puede ser despreciado. En este ejemplo, cuyo fin es exponer un método de cálculo, no se tienen en cuenta los valores propios del fenómeno físico.

La fórmula obtenida, conociendo los valores de A, B y C (constantes que dependen de las características del recipiente), suministra la temperatura T de equilibrio a la que se mantiene el agua bajo el efecto combinado de la energía de entrada y la de salida. En dicha fórmula, la temperatura T no viene expresada en la escala de grados centígrados a la que estamos acostumbrados, sino en grados Kelvin (los grados Kelvin se calculan sumando 273,2 a los grados centígrados; en esta escala el agua hierve a 100 + 273,2 = 373,2 grados Kelvin, mientras que el hielo funde a 273,2 grados Kelvin = 0° centígrados).

Los grados Kelvin se expresan con números altos (por ejemplo, 373,2 para el agua en ebullición); para comodidad de cálculo se suele dividir el vapor por 1.000 (tendremos: agua en

TEST 5

1 / El símbolo ⬡ representa:
a: una decisión; b: el desarrollo de un cálculo; c: el final del programa.

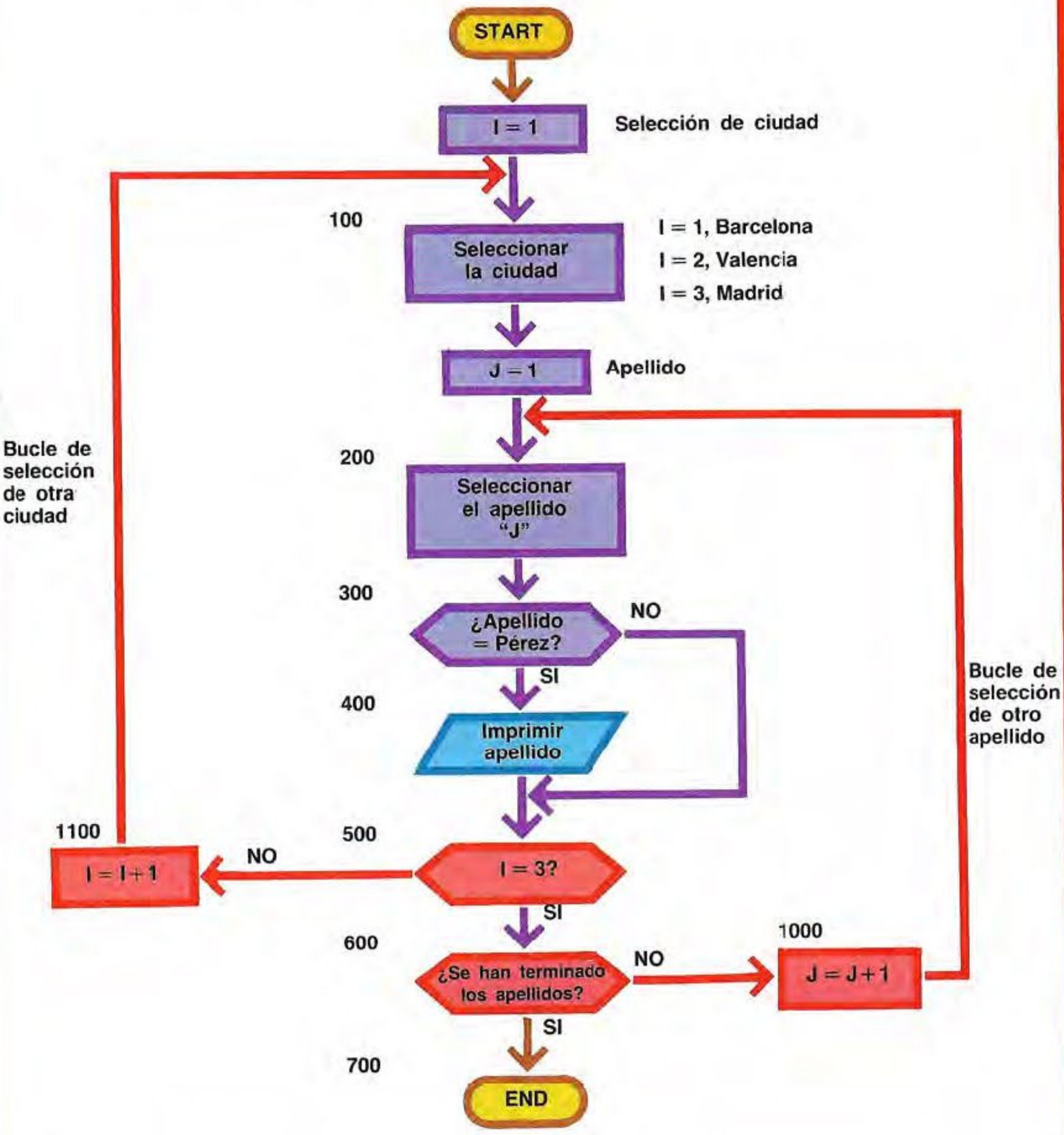
2 / Dibujar los símbolos que representan las siguientes acciones:
a: acción genérica; b: cálculo; c: impresión.

3 / ¿Para qué sirve el símbolo: ○ ?

4 / ¿Qué es un flag?:
a: un indicador genérico; b: un contador; c: un registro de la CPU.

5 / Dibujar el diagrama de flujo de un programa para calcular porcentajes, con control en la introducción, de manera que el porcentaje no sea superior a 100.

6 / El diagrama de flujo siguiente contiene un error. Determinar cuál es y elaborar la forma correcta del diagrama. El problema es el siguiente: hay tres grupos de direcciones; el primero relativo a Barcelona, el segundo a Valencia y el tercero a Madrid. Cada grupo contiene 10 direcciones. Se desea seleccionar todas las direcciones de los señores Pérez (hacen falta dos bucles, uno para seleccionar la ciudad, el otro para el apellido).



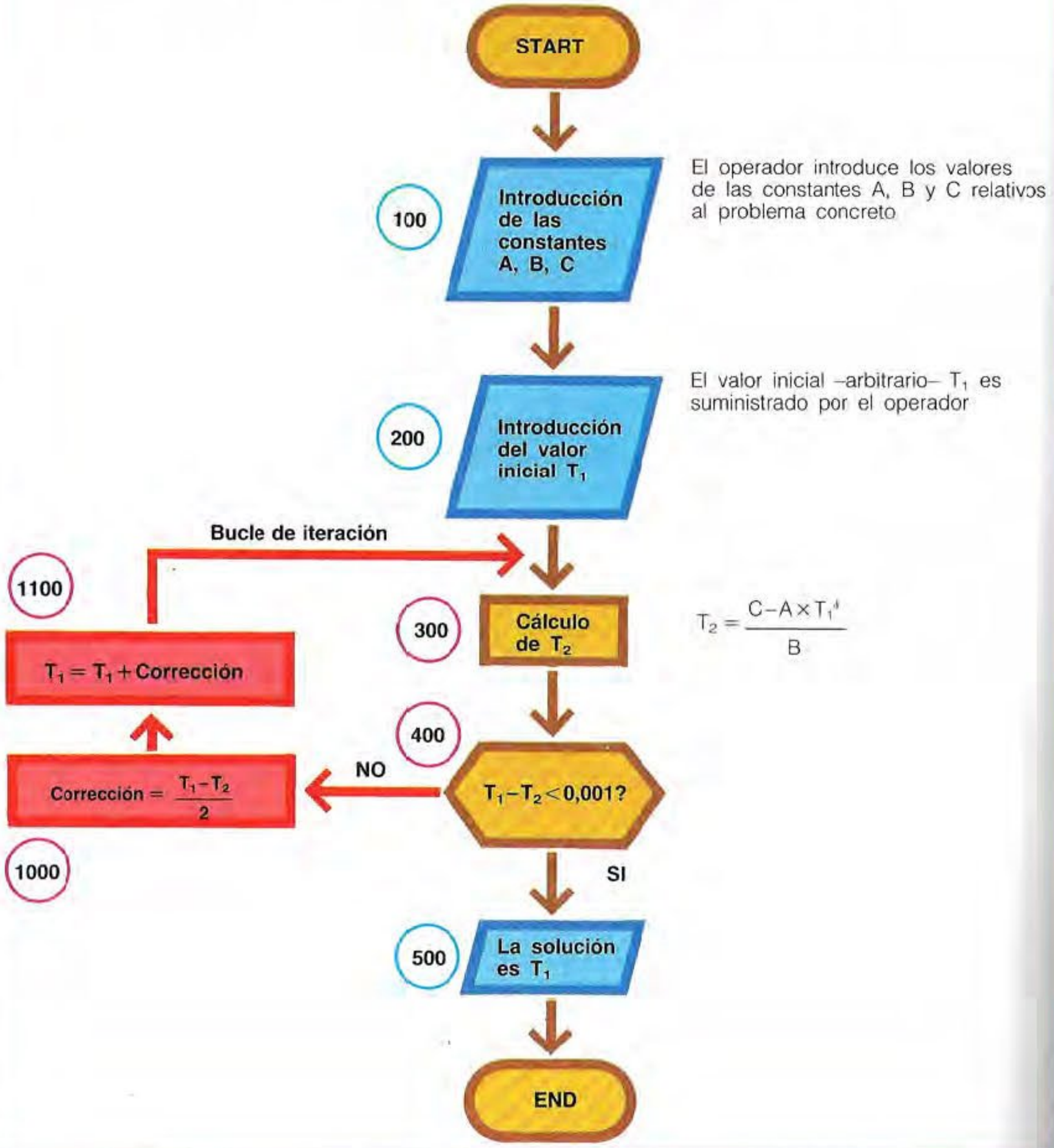
Las soluciones, en las págs. 214 y 215.

ebullición = 0,3732; hielo = 0,2732; etc); de esta manera, todos los cálculos se efectúan con números pequeños. Así por ejemplo, la temperatura de 373,2 K a la 4ª potencia (T^4) es 19.398.428.000, mientras que el equivalente dividido por 1.000 (o sea, 0,3732) da 0,0193984, que se puede redondear como 0,0194. Naturalmente, al final de los cálculos hay que multiplicar el resultado por 1.000 (este método también resulta cómodo para los valores concretos de las constantes que aparecen en los cálculos térmicos).

Reescribamos la fórmula del equilibrio térmico:

C calor cedido por el sol	=	$A \cdot T^4$ pérdidas por radiación	+	$B \cdot T$ pérdidas por conducción

DIAGRAMA DE FLUJO PARA LA RESOLUCION CON EL METODO ITERATIVO DE ECUACIONES DEL TIPO $A \times T^4 + B \times T = C$



La ecuación puede escribirse también de la siguiente forma:

$$T = \frac{C - A \cdot T^4}{B}$$

La solución T (temperatura a la que llega el recipiente) se obtiene con el siguiente procedimiento iterativo:

1 / En el término $A \cdot T^4$ se da un valor arbitrario a T (por ej., T_1) con el que se efectúan los cálculos indicados en el segundo miembro de la ecuación. Al finalizar los cálculos, tenemos en el primer miembro un nuevo valor de T, que llamaremos T_2:

$$T_2 = \frac{C - A \cdot T_1^4}{B}$$

2 / Si los dos valores T_1 y T_2 difieren poco (por ejemplo 0,001), se ha encontrado la solución.

3 / En caso contrario se calcula un valor de corrección para T igual a la media de los valores T_1 y T_2:

$$\text{corrección} = \frac{T_1 - T_2}{2}$$

4 / Esta corrección se aplica al valor inicial (arbitrario) T_1, valor que se convierte en T_1 + corrección.

5 / Con este nuevo valor de la temperatura se vuelve a partir del punto 1.

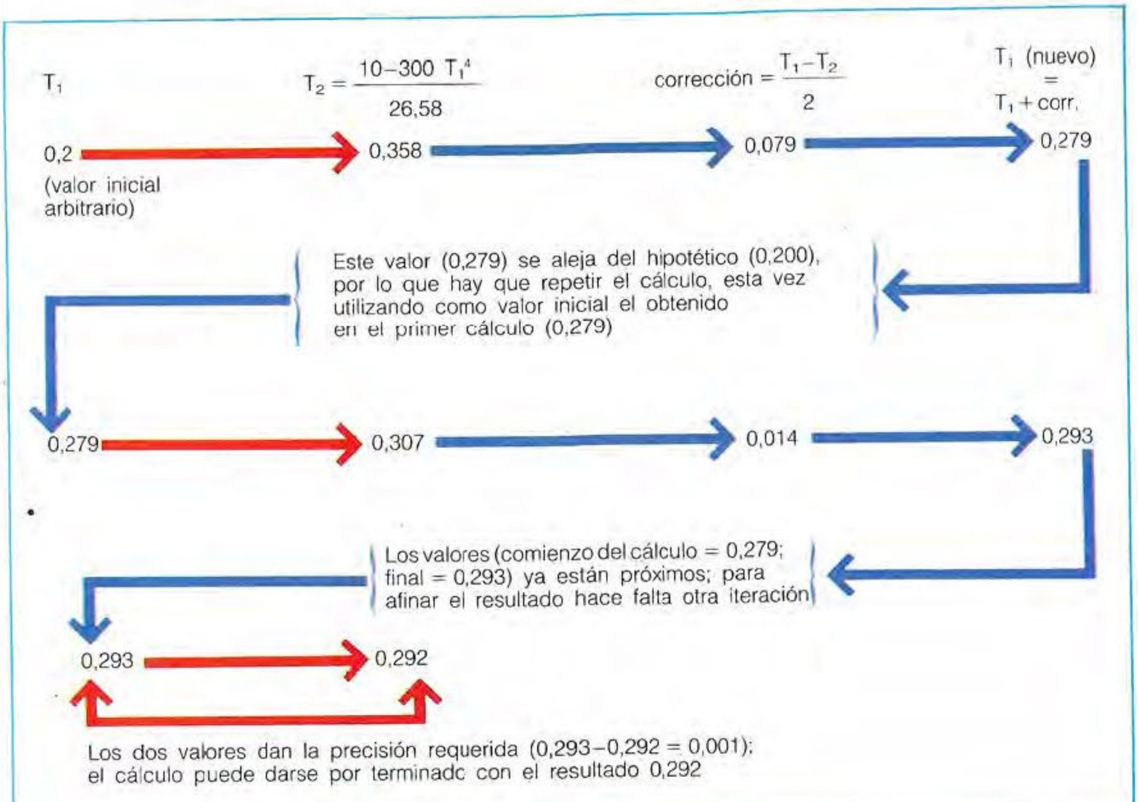
En el gráfico de la pág. 208 se muestra el diagrama de flujo de este método. En la tabla de abajo se realiza un cálculo adoptando los valores A = 300, B = 26,58 y C = 10, con los que la ecuación se convierte en:

$$300 \cdot T^4 + 26{,}58 \cdot T = 10$$

Programación estructurada

Los métodos que hemos visto para el análisis del problema, para su descomposición en un determinado número de problemas más simples y, finalmente, para la preparación de los diagramas de flujo, permiten definir la estructura lógica necesaria para resolver la mayoría de las aplicaciones prácticas. Sin embargo, esta metodología, aunque válida, es muy empírica y no resulta fácilmente aplicable a problemas más complejos.

El método HIPO y la preparación de los diagramas de flujo son eficazmente aplicables en la



programación de micro y miniordenadores. En los sistemas más complejos hay que adoptar técnicas más rigurosas, que establezcan reglas lo suficientemente estrictas como para permitir el trabajo en equipo.

En la preparación de un programa para micro o miniordenador trabaja, normalmente, una sola persona; por tanto, no hace falta conectar entre sí partes escritas por distintos programadores. Se puede, pues, aceptar como metodología de trabajo el sentido común, más que la rígida observancia de reglas y estructuras preestablecidas. Al tratar problemas más complejos aumenta el número de personas que contribuye a la tarea, por lo que aumenta la necesidad de claridad en la documentación y de una forma de «estandarización» de las metodologías a seguir. Además, hay que adoptar técnicas que permitan reducir las posibilidades de error y ahorrar tiempo. La más moderna y sofisticada de estas técnicas es la **programación estructurada**. Se trata de una metodología compleja que sólo puede ser útil en las aplicaciones con grandes sistemas (main frame o miniordenadores de los de mayor envergadura). En este apartado se describen algunos de los principales aspectos de esta técnica de programación.

La programación estructurada viene a ser un método de preparación de los diagramas de flujo basado en cinco funciones principales.

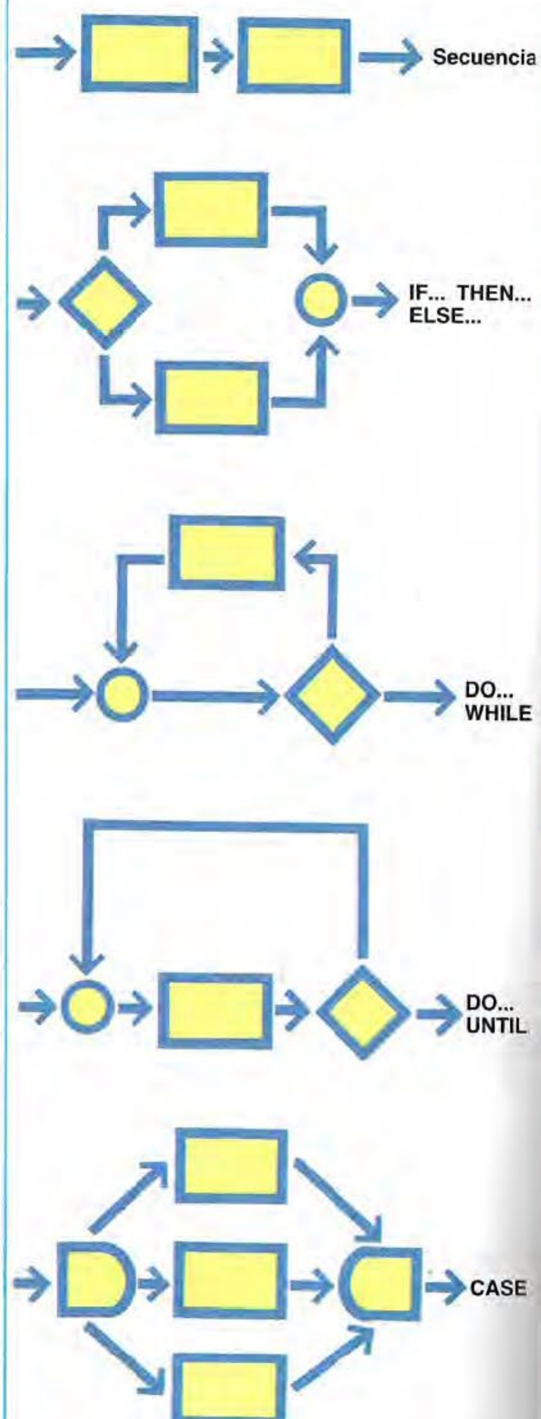
En la diagramación normal hay un símbolo por cada posible acción a realizar en el seno de un programa. Puesto que estos símbolos describen acciones elementales, los diagramas basados en ellos son muy detallados. En la programación estructurada no existen símbolos elementales, sino sólo símbolos compuestos que expresan acciones complejas. Así, gran parte de los diagramas de flujo usados para describir un bucle con ruptura de código pueden expresarse con un único símbolo complejo.

Naturalmente, un sistema tal sólo puede utilizarse si se dispone, en la fase de escritura del programa, de un lenguaje adecuado; es inútil preparar un diagrama con metodologías sintéticas si luego, en la traducción en instrucciones, hay que desarrollar los detalles.

Los lenguajes más comunes de los micro y miniordenadores (Basic, Cobol, Fortran) son poco adecuados para la programación estructurada, salvo en algunos casos particulares con accesorios no estandarizados.

Son más adecuados lenguajes como el **PL/I**, el

SIMBOLOGIA DE LA PROGRAMACION ESTRUCTURADA

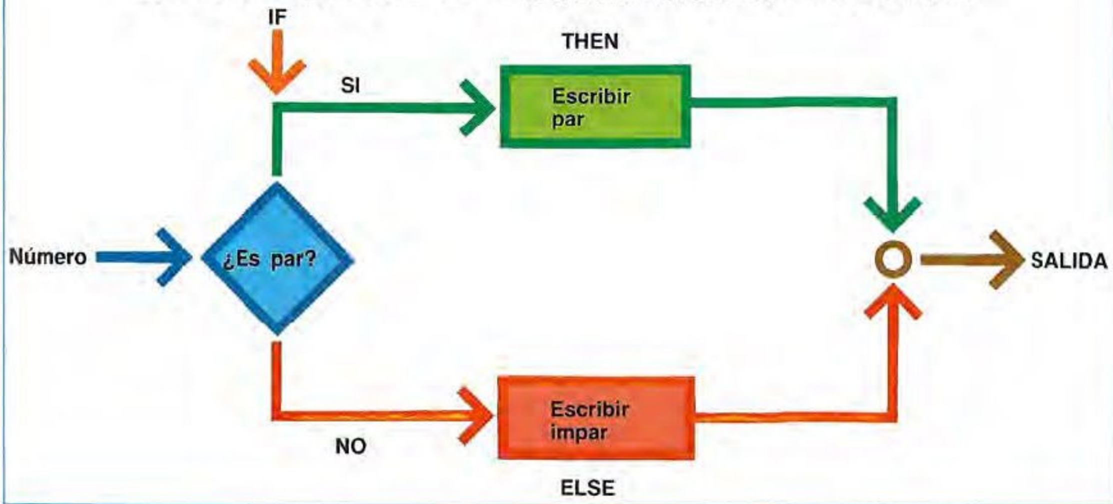


PL/C, el **PASCAL** o el **ALGOL**. Estos lenguajes son casi desconocidos en las aplicaciones de escasa complejidad; relativamente recientes, se consideran «interesantes» porque permiten un notable ahorro de tiempo. No hay que excluir la posibilidad de que en un futuro próximo tengan una notable difusión (el PL/I y el PASCAL ya se utilizan en micro y miniordenadores).
Las principales funciones de la programación estructurada se esquematizan en el gráfico de la pág. 210 y se describen a continuación.

Secuencia

El símbolo gráfico expresa claramente la función: una serie de operaciones (de cualquier tipo) a efectuar en serie.

DIAGRAMA DE FLUJO DE UNA DECISION REESCRITA CON LA SIMBOLOGIA DE LA PROGRAMACION ESTRUCTURADA



IF... THEN... ELSE...

Es un símbolo compuesto. Comienza con una decisión (test de condición); según el resultado, son posibles dos caminos, uno si se cumple la condición, el otro en caso contrario. La salida es común.
En el gráfico superior vemos una aplicación. En la entrada se suministra al bloque un número y se pide la impresión del mensaje PAR si el número es par, y en caso contrario el mensaje ha de ser IMPAR.
Interpretando literalmente las frases contenidas en este bloque, tenemos:

IF «Número es par» THEN «Escribir par»
ELSE «Escribir impar»

Se trata, en esencia, de una forma muy parecida al símbolo de decisión similar.

DO... WHILE y DO... UNTIL

Los dos bloques son similares y contienen un «test» (símbolo de decisión) y una acción genérica. En el primero, tras la entrada, se efectúa el test; si la respuesta es positiva, se efectúa la acción y se vuelve a entrar en el test. La salida se obtiene cuando el test da resultado negativo. En el segundo, la acción precede al test. En esencia, son dos formas compactas de bucle.

CASE

Este símbolo no tiene una correspondencia inmediata en las simbologías de los diagramas de flujo. Su significado es el de una comparación múltiple; para obtener el mismo resultado con símbolos de decisión, hay que utilizar tantos como posibles caminos a seguir. La aplicación de este símbolo, o de su equivalente realizado con los símbolos de decisión normales, se aclara mejor con un ejemplo. Los impuestos sobre el sueldo dependen de su importe. Supongamos que hay sólo cuatro niveles:

BLOQUES DO... WHILE / DO... UNTIL

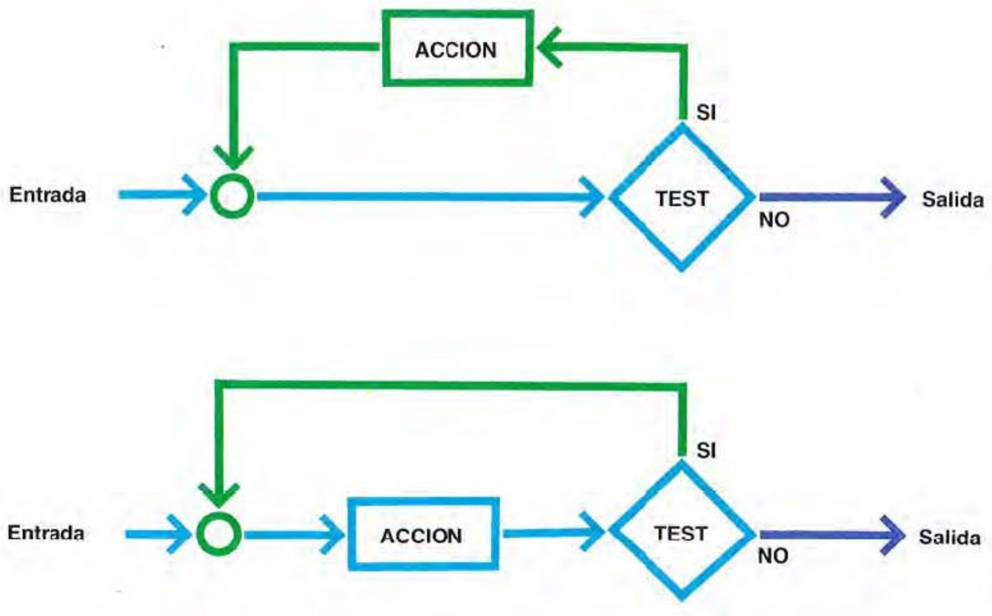
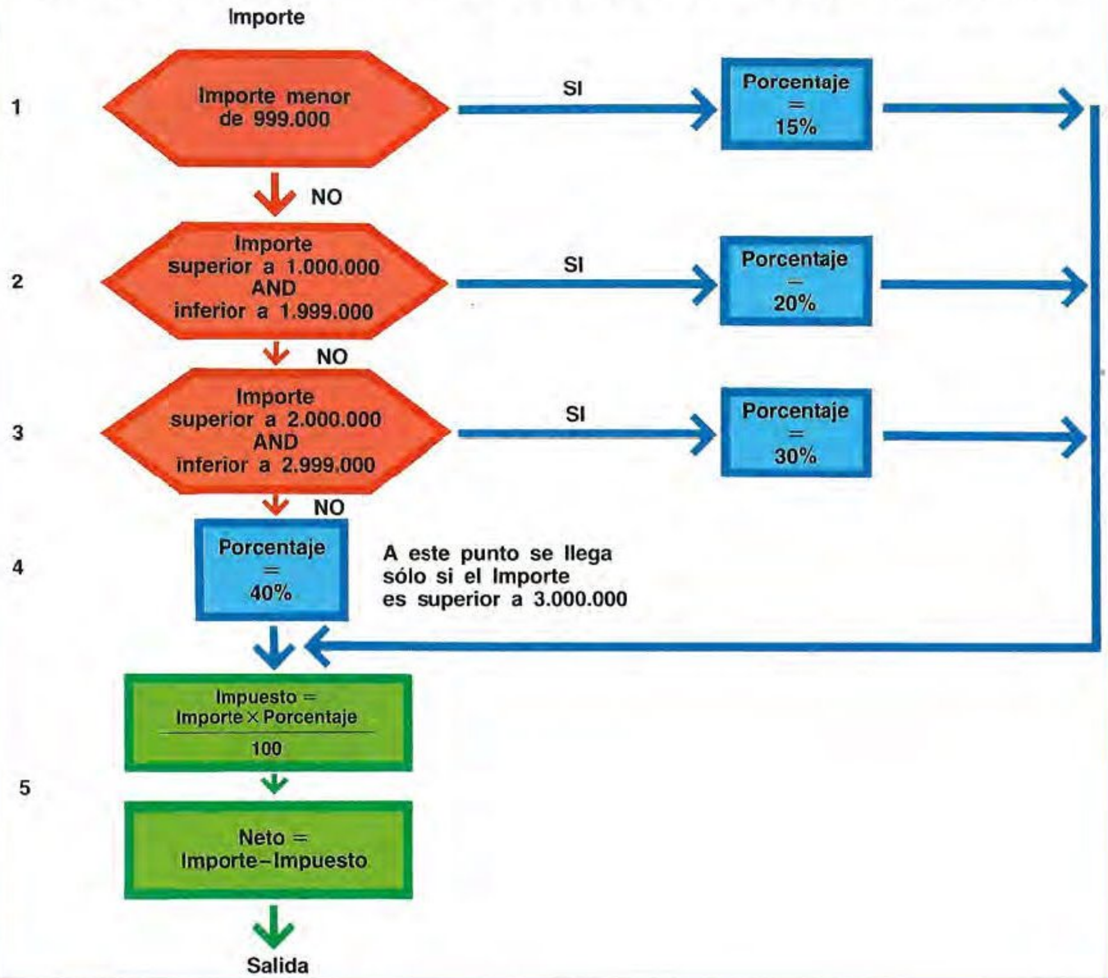


DIAGRAMA DE FLUJO PARA LA SELECCION DE LA CUOTA IMPOSITIVA



Sueldo	
hasta 999.000	porcentaje = 15%
de 1.000.000 a 1.999.000	porcentaje = 20%
de 2.000.000 a 2.999.000	porcentaje = 30%
más de 3.000.000	porcentaje = 40%

El problema consiste en escribir un programa capaz de decidir qué cuota aplicar, y de calcular el importe del impuesto y del sueldo neto.

En el gráfico superior se muestra el diagrama de flujo que resuelve el problema con la simbología normal.

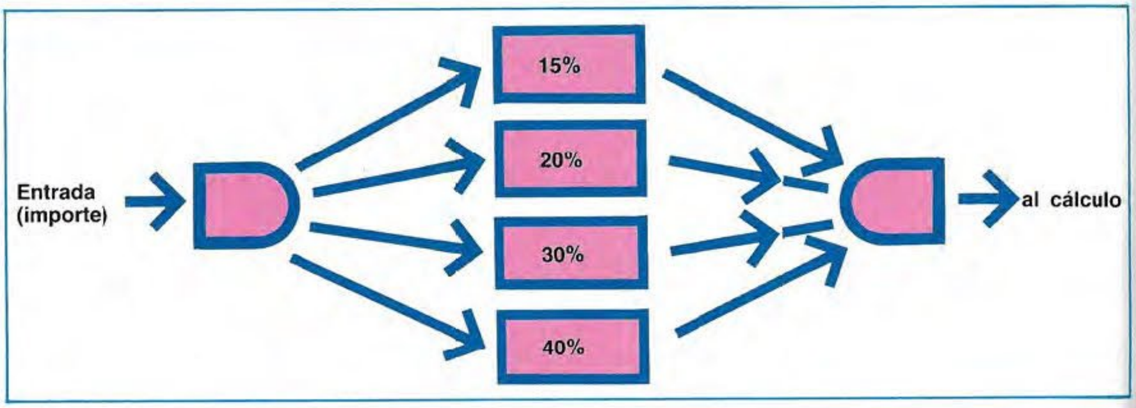
La decisión **1** controla que el importe no sea superior a 999.000; en este caso se aplica el 15%; una vez hallado el tanto por ciento, el programa salta directamente a los cálculos (punto **5**).

En el caso de que la respuesta del bloque **1** sea negativa, se pasa al test **2**. En este punto, si el importe está contenido en la segunda franja (entre 1.000.000 y 1.999.000), se selecciona la cuota del 20% y se pasa al cálculo.

Obsérvese el uso del operador AND para expresar dos condiciones que deben cumplirse simultáneamente (importe superior a 1.000.000 e inferior a 1.999.000, es decir, comprendido entre 1.000.000 y 1.999.000). El punto **4** no precisa control alguno, ya que si en el punto **3** el importe no es inferior a 2.999.000, hay que aplicar el porcentaje máximo.

Utilizando la forma CASE se puede expresar esta lógica de forma más sintética, puesto que el CASE puede considerarse como los tres símbolos de decisión reunidos. En el gráfico de la pág. 214 se muestra el esquema de principio.

Al programar en Basic (o Fortran) no se dispone de instrucciones que permitan el empleo de lógicas tan compactas.



SOLUCIONES DEL TEST 5

1 / a: el símbolo representa una decisión. El desarrollo de un cálculo se indica con un rectángulo como símbolo genérico, escribiendo en su interior el cálculo a efectuar. El final de un programa se indica mediante un óvalo con la palabra END.

2 / a:  el símbolo para la acción genérica se utiliza también para los cálculos, por lo tanto las acciones a y b se representan con este símbolo.

b:  las operaciones I/O se indican ambas con el mismo símbolo. En el interior del símbolo, se escribe de cuál de las dos se trata.

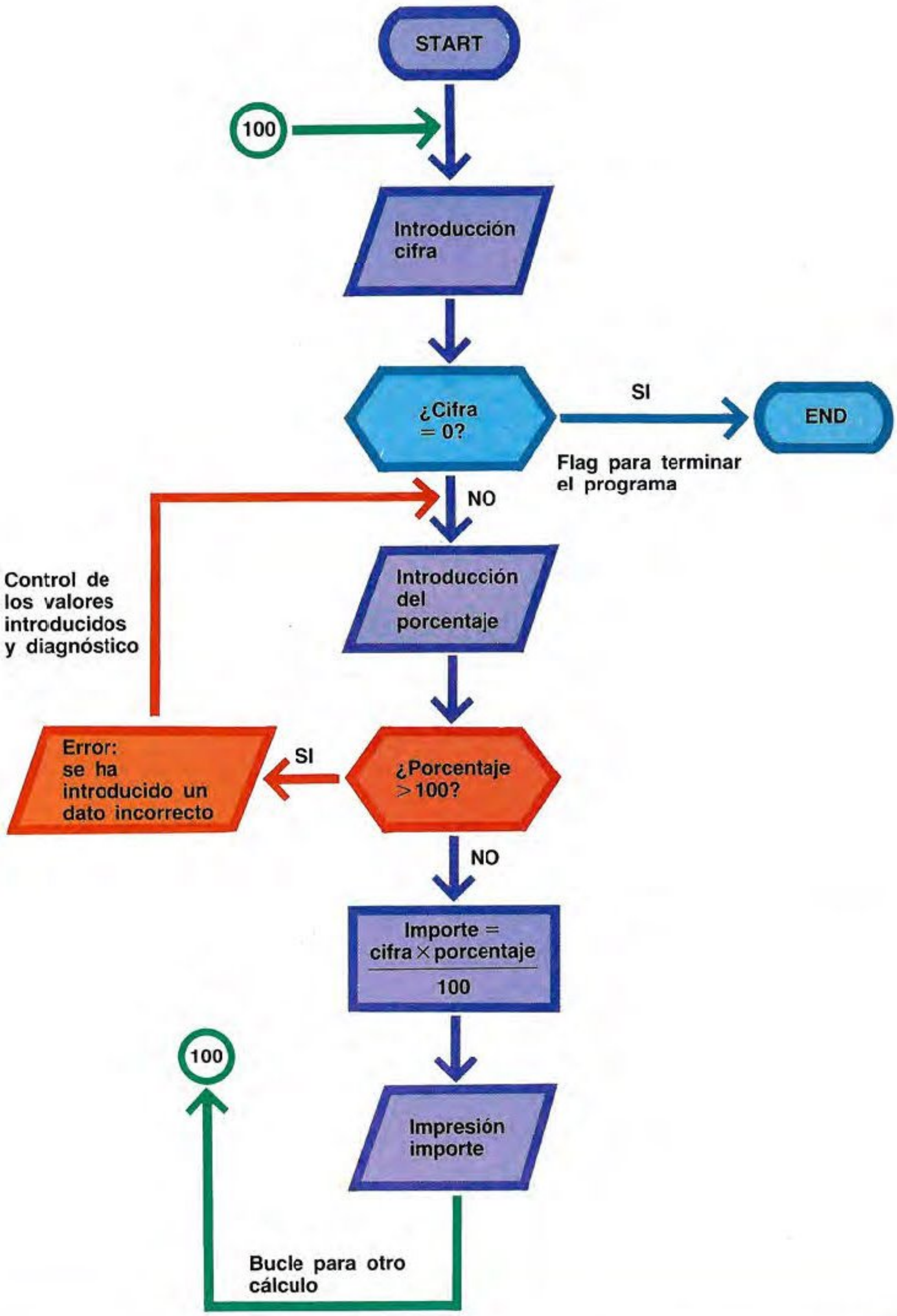
3 / Este símbolo (conector) se utiliza para indicar puntos del diagrama de flujo a los que se hace referencia en partes alejadas (en el dibujo), para conectar con las cuales resultaría incómodo o complicado utilizar una línea de flujo continua.

4 / a: el flag es un indicador que, con su estado, señala si un determinado hecho se ha verificado o no.

5 / El análisis del problema se puede sintetizar de la forma siguiente:
— introducción del número del que hay que calcular el porcentaje;
— comprobar que dicho importe no sea ¢ (el valor ¢ puede utilizarse como indicador -flag- que permite interrumpir el programa);
— introducción del valor del porcentaje y control del mismo;
— cálculo: importe % = cifra x porcentaje/100;
El diagrama de flujo se muestra en el gráfico de la página contigua.

6 / El error consiste en la presencia de dos bucles (I y J) que se cortan. El bucle con índice I (que comienza en el bloque 100) ha de cerrarse después del índice J (que comienza en 200), de manera que no hay intersecciones. La visión clara del error se obtiene llevando el bucle que pasa por el bloque 1.100 (a la izquierda en el gráfico de

la pág. 207) al mismo lado del bucle 1.000 (a la derecha); se ve claramente que las dos líneas de retorno se cortan, y eso no está permitido. Se corrige cambiando entre sí los dos bloques 500 y 600 de cierre de los bucles.



Las máquinas que hablan

Las capacidades que la informática ha desarrollado en los últimos años permiten ya pensar seriamente en el intercambio de información verbal entre el hombre y la máquina.

Desde el punto de vista práctico, hay dos problemas distintos: ante todo, el de construir máquinas capaces de reconocer preguntas verbales y contestarlas; luego, el de convertir las respuestas en un discurso articulado. Este último objetivo –hacer una máquina parlante– es más sencillo que el primero.

Los modernos dispositivos electrónicos parlantes se basan en dos principios fundamentales. El primero, la «síntesis verbal concatenada», utiliza un ordenador para almacenar un «diccionario» de sílabas pronunciadas en la lengua deseada, con todas las combinaciones posibles de vocales-consonantes y de consonantes-vocales. Para formar una palabra, el ordenador extrae de su memoria las sílabas necesarias y las une en el orden correcto («or-de-na-dor», por ejemplo). Luego, añade las pausas y la entonación necesarias para que su «producto» sea inteligible. El método de la síntesis concatenada es muy rígido y produce discursos de baja calidad; sin embargo, trabaja con rapidez y es de construcción bastante económica.

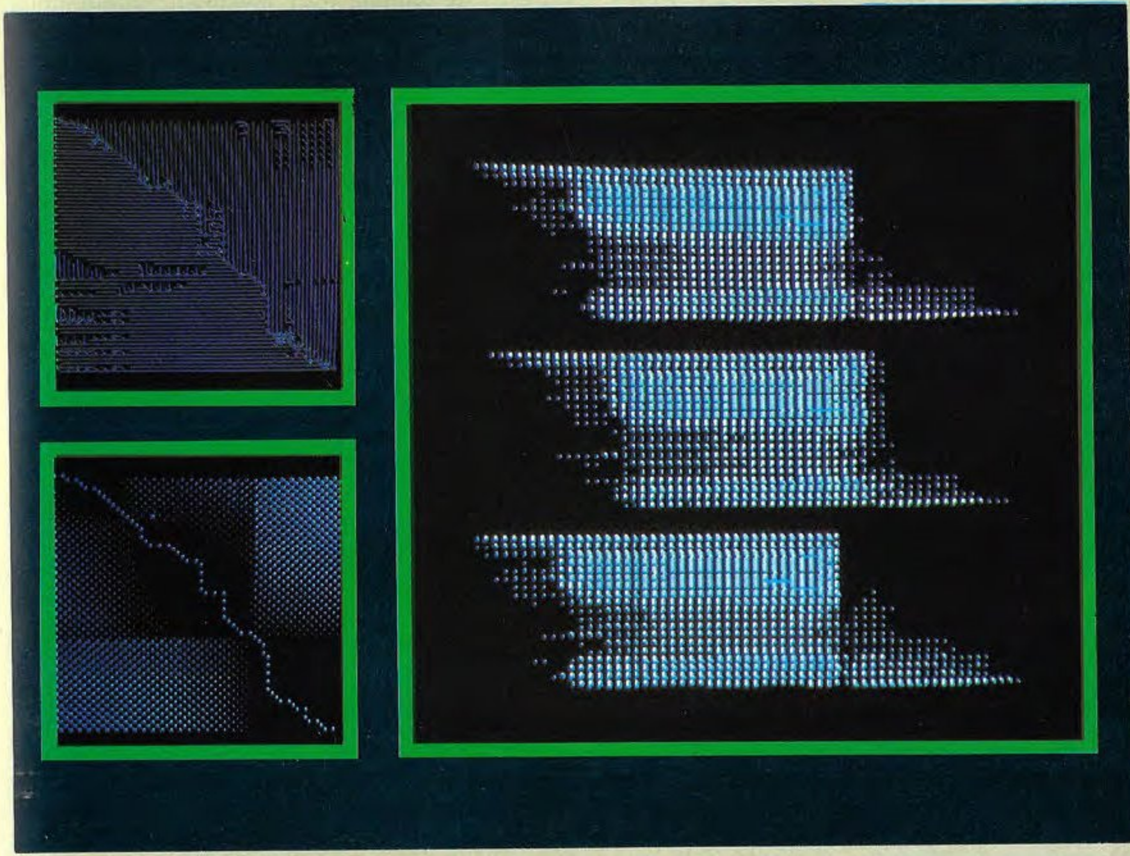
El otro método, la «síntesis por reglas», no almacena en la memoria sonidos o combinaciones de sonidos, sino más bien una guía de «reglas» electrónicas, sacadas de un análisis detallado del lenguaje natural, necesarias para generar esos sonidos. La síntesis por reglas produce un discurso de mejor calidad; sin embargo, debido al hecho de que hay que elaborar muchas reglas para producir incluso una breve concatenación de sonidos articulados, la máquina emplea un tiempo considerable para sintetizar el lenguaje. Diversos tipos de máquinas parlantes que utilizan la «síntesis verbal concatenada» son ya habituales, sobre todo en Estados Unidos. Hay calculadoras parlantes que pueden pronunciar cada número tecleado y luego leer en voz alta el resultado; además, hay pequeños dispositivos parlantes, poco mayores que una calculadora, que ayudan a enseñar a hablar a los niños. Otra pequeña máquina no sólo juega muy bien al ajedrez, sino que anuncia verbalmente las jugadas que realiza.

Una máquina «parlante» es la parte fundamental de un sistema usado en Estados Unidos por la Bell Telephone Company para aumentar la eficacia y reducir los costos de algunos servicios atendidos por operadores humanos. En el teléfono se inserta un teclado similar al de una máquina de escribir, pero más pequeño. Para obtener el número de un abonado, se conecta con el servicio de información y se teclea su nombre y dirección. El ordenador, al otro extremo de la línea, busca en su memoria, encuentra el número pedido y, con la «voz» de la máquina, transmite la respuesta por teléfono.

La siguiente etapa de desarrollo, que prevé la eliminación del teclado y la solicitud de la información directamente a la máquina, es un paso mucho más difícil.

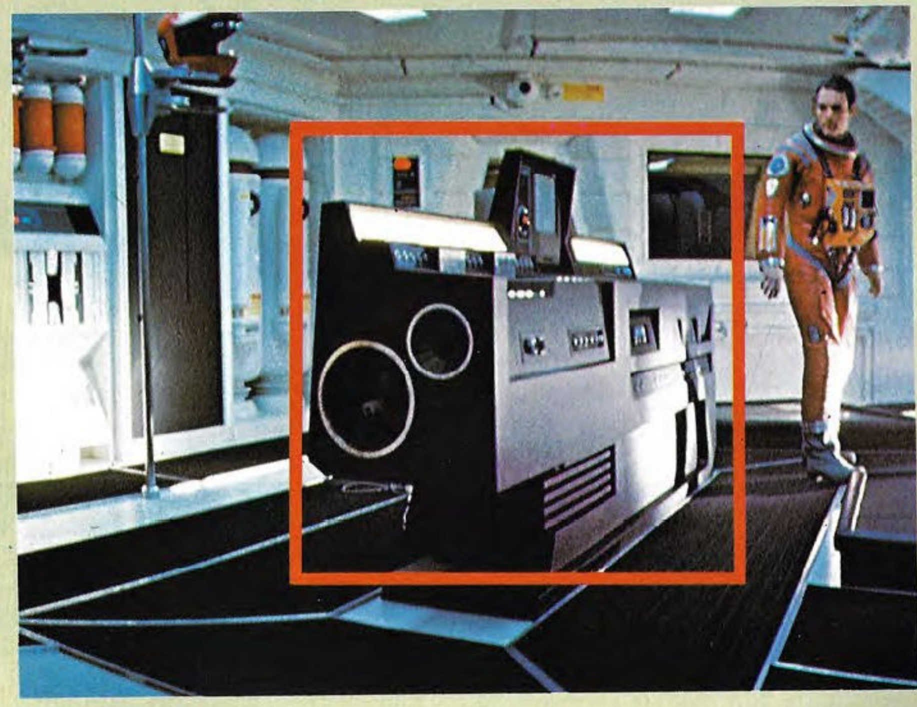
Un instrumento fundamental para los científicos que intentan poner a punto un sistema mediante el cual las máquinas puedan reconocer el lenguaje articulado, es el «espectrógrafo acústico», inventado a comienzos de la Segunda Guerra Mundial. Se trata de un dispositivo electrónico que analiza las ondas acústicas y reproduce un «cuadro» gráfico permanente de las mismas. La importancia de los cuadros espectrográficos estriba en que pueden almacenarse (de forma digital) en la memoria de un ordenador. En teoría, sería fácil para éste reconocer una determinada palabra confrontando el espectrograma de cada sonido recibido con los datos espectrográficos archivados en su memoria. El problema está en que no hay dos personas que pronuncien la misma palabra exactamente de la misma forma: palabras idénticas, pronunciadas por sujetos diferentes, producen cuadros espectrográficos decididamente distintos que, obviamente, crean confusión.

La forma más eficaz encontrada hasta el momento para superar esta dificultad, es la de «adiestrar» a la máquina. Un método, pues, muy similar al utilizado para enseñar a los niños a hablar. El «adiestramiento» comienza con un grupo de unas cien personas, todas con el mismo acento regional, cada una de las cuales pronuncia todas las palabras que se desea que la máquina reconozca. Los diversos espectrogramas registrados son introducidos luego en el ordenador, que analiza los cuadros gráficos de una misma palabra y elabora un cuadro «intermedio», «silueta», que memoriza para futuras comparaciones. Otra tarea del ordenador consiste en evaluar la desviación de la «silueta» del espectrograma de un determinado sonido emitido por una persona.



MC/Theo Bergström

La palabra «sum» analizada por el ordenador. Los diagramas de la izquierda representan dos versiones de esta palabra pronunciada por la misma persona. La máquina hará una síntesis.



Kobal Collection MGM Inc.

Hal, el ordenador de la película «2001, una odisea del espacio», es un típico ejemplo de máquina parlante de ciencia ficción.

A partir de ese momento, los modelos espectrográficos de las palabras dirigidas a la máquina serán comparados con las distintas «siluetas» memorizadas, teniendo en cuenta la variación teórica. Las «siluetas» pueden ser «estiradas» (deformadas) de forma que puedan asociarse a palabras pronunciadas a distintas velocidades: cuando una «silueta» encaja con una palabra recibida, la máquina reconoce la palabra.

Los sistemas que operan de esta forma y que, según se dice, reconocen palabras pronunciadas por cualquier orador que hable la misma lengua, pueden obtener índices de éxito de un 80-90%; si, además, los oradores son los mismos que han «adiestrado» a la máquina, el éxito puede ser de un 98%.

Este principio hace posible la construcción de máquinas capaces de comprender órdenes sencillas y obrar en consecuencia. Un ejemplo podría ser una máquina de bebidas a la que se pudiera decir «té», «leche», «azúcar» para que suministrara, tras la elaboración de la orden, una taza de té con leche y azúcar. Pero el paso siguiente, es decir, de la comprensión de palabras aisladas a la de discursos articulados enteros, o sea de frases y proposiciones completas, es tan arduo que pocos investigadores se han dedicado a ello, ya que los problemas que plantea son enormes.

En 1970, por ejemplo, el Departamento de Defensa de Estados Unidos pidió a un grupo de científicos que especificaran lo que, razonablemente, cabía esperar de un sistema de reconocimiento verbal. Las conclusiones fueron que un sistema tan sólo podría utilizar un vocabulario limitado a mil palabras (los sujetos con una educación escolar media-superior poseen un vocabulario que incluye entre 5.000 y 10.000 términos), y estaría dotado de una capacidad de sintaxis (estructuración de frases y proposiciones) muy simple. Sería, por otra parte, necesario que la tarea del sistema fuera muy específica, tanto en la manipulación de los datos como en la programación del ordenador. El Input *(introducción) verbal debería ser continuo, y no fragmentado en sílabas, pues eso «engañaría» a la máquina; ésta podría aceptar discursos de un amplio número de operadores, pero todos con el mismo acento regional, que tendrían que hablar con voz clara ante un micrófono de alta calidad y en una habitación sin ruidos.*

A finales de los setenta, los sistemas de reconocimiento verbal (o mejor dicho, de reconocimiento de palabras aisladas) no respondían ni siquiera mínimamente a estos requisitos aparentemente modestos. Los vocabularios de las máquinas eran extremadamente restringidos: en lugar de los 1.000 términos recomendados, dos de los mejores sistemas podían utilizar un vocabulario de 39 «palabras» (las 26 letras del alfabeto, las diez cifras de 0 a 9 y tres palabras de comando) o un vocabulario de 54 palabras en la programación verbal del ordenador.

En 1980, un equipo de investigadores de IBM anunció que había alcanzado la meta de las 1.000 palabras. Dichos investigadores utilizaron un ordenador IBM 370 para transponer por escrito frases verbales. Sin embargo, también en este caso hay inconvenientes: ante todo, cada operador que utilice la máquina ha de dedicar dos horas a decir por un micrófono 900 frases de «entrenamiento»; además, el proceso de elaboración es muy lento (una frase que requiere 30 segundos para ser formulada verbalmente por el operador, puede tardar 100 minutos en ser transcrita), y, finalmente, las posibilidades de error son elevadas.

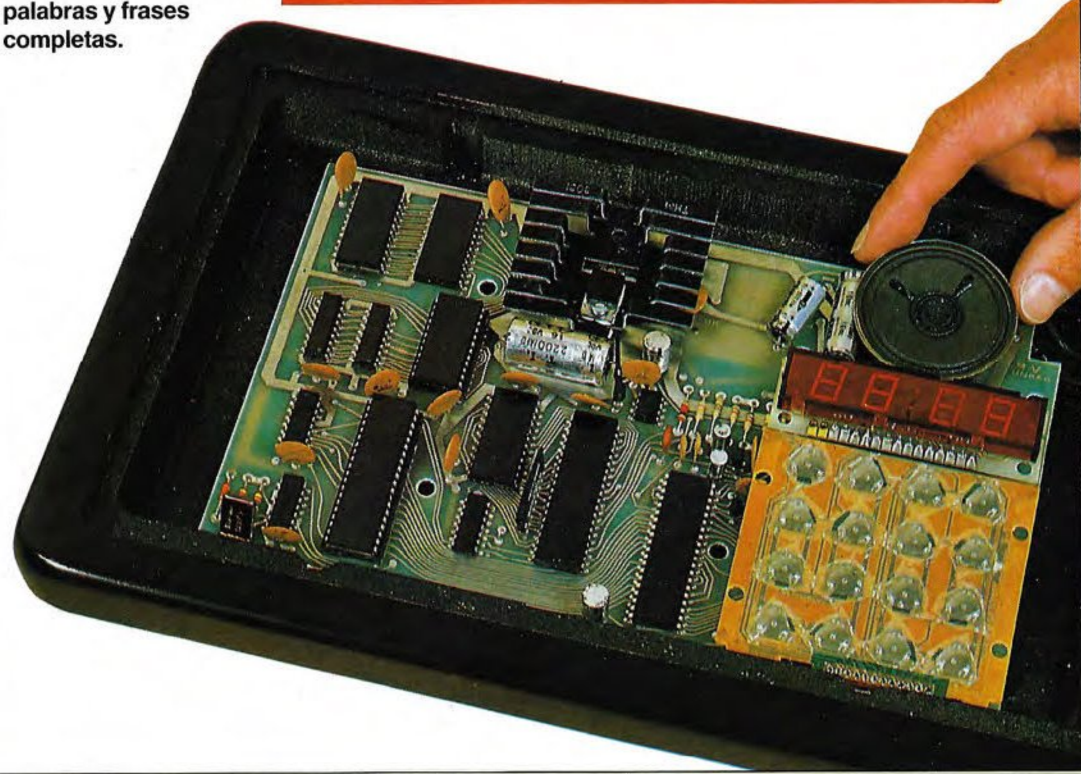
Existen problemas en el reconocimiento del lenguaje que, obviamente, la Comisión de 1970 no alcanzó a prever.

En primer lugar, incluso un fragmento verbal muy corto (una conversación, por ejemplo, o una transmisión de noticias) contiene centenares de pequeñas partículas verbales, o fonemas, que la máquina ha de identificar minuciosamente y luego seleccionar, estableciendo dónde termina una palabra y comienza la siguiente; un problema complicado por el hecho de que la mayoría de las personas habla de prisa, diciendo las palabras «todas juntas».

En segundo lugar, incluso una máquina capaz de escindir eficazmente una frase en sus palabras integrantes tendría que afrontar problemas tales como la identificación de palabras que suenan igual, pero tienen significados distintos. Por otra parte, se perdería mucho tiempo en buscar en la mole de un diccionario cada sucesión de sonidos, cada palabra, cada combinación de palabras y así sucesivamente, hasta encontrar la correcta. El sistema podría funcionar bien, pero con extremada lentitud. Otra solución, que podría ser mucho más económica, sería la de enseñarle a la máquina no todo el lenguaje, sino sus reglas, o sea cómo se construyen las frases y períodos partiendo de la base de las palabras sueltas.

A la derecha y abajo. He aquí una máquina que no sólo juega muy bien al ajedrez, sino que también «dice» las jugadas en el mismo momento en que las hace. El sistema se sirve de la denominada «síntesis concatenada»: partículas de sonidos verbales, almacenadas en la memoria, se conectan electrónicamente en la secuencia adecuada para formar palabras y frases completas.





Computer Games Ltd

Se trataría, pues, de saber qué tipo de palabras puede ir tras un sustantivo, cuál detrás de un verbo, cuál al comienzo de una frase, cuál al final, y así sucesivamente.

Sin embargo, existe el problema de que ninguna lengua posee reglas gramaticales absolutamente rígidas. Por ejemplo, se podría enseñar a la máquina la regla de que, en castellano, ninguna frase termina con el artículo «el». Sin embargo, una regla tal sería válida 999 veces de cada mil, pero no siempre, como lo demuestra la frase anterior.

En cualquier caso, lo cierto es que los actuales sistemas de reconocimiento de lenguaje tendrán que ser perfeccionados, y este perfeccionamiento tal vez se consiga empleando métodos totalmente distintos de los que se están investigando actualmente.

Archivo de datos

Los datos elaborados con el ordenador tienen la característica de no poder guardarse de manera estable en su memoria central. Efectivamente, la característica fundamental de los datos presentes en la memoria central del ordenador es la de su temporalidad de uso. Los datos permanecen el tiempo que dura la elaboración en curso; posteriormente son sustituidos por otros datos, correspondientes a una nueva elaboración. Se plantea, de este modo, la necesidad de memorizar aquellos datos relativos a una elaboración antes de que una nueva utilización de la máquina venga a destruir el contenido de la memoria.

Considerando el gran número de datos que el ordenador puede manejar y producir, y teniendo en cuenta la velocidad con que se elaboran, es necesario memorizar los datos de salida en soportes que sean capaces de contenerlos en gran número.

Los datos han de ser archivados y memorizados de forma tal que puedan ser utilizados como datos de Input en nuevas elaboraciones.

Teniendo en cuenta, además, que el ordenador pierde el contenido de la memoria central al cortarse el suministro eléctrico, es evidente la necesidad de archivar los datos en soportes más fiables, que conserven su contenido incluso en ausencia de suministro eléctrico y durante un tiempo considerable (del orden de años).

Los sueldos de los empleados de una empresa, por ejemplo, se calculan una vez al mes y, por lo tanto, los datos han de estar disponibles en un período fijo del mes (pongamos los días 20 y 21). Es de todo punto inconcebible que se tenga que abarrotar la memoria central del ordenador, por grande que sea, con esos millones de caracteres los demás días del mes; por otra parte, una vez calculada la retribución del señor Pérez, lo lógico es que el resultado del cálculo se archive en otra parte, dejando libre el correspondiente espacio de la memoria central para los datos del cálculo de la retribución del señor García.

Este ejemplo hace evidente, pues, la necesidad que existe de valerse de otros recursos del ordenador; más concretamente se trata de la posibilidad de utilización de los soportes que permiten archivar datos.

Estos soportes externos se denominan memorias auxiliares, masivas o de masa.

Memorias de masa

Dejemos de lado la posibilidad de archivar datos en fichas o cintas perforadas, puesto que son soportes ya superados o utilizados sólo en aplicaciones especiales, y centrémonos en los soportes que permiten el registro magnético de los datos. Con respecto a ellos cabe hacer algunas observaciones de índole general.

La lectura y el registro se realizan mediante aparatos adecuados (unidades de cinta o de disco) provistos de cabezas de lectura y escritura. El acceso a las distintas porciones del soporte magnético se efectúa mediante movimientos mecánicos de la memoria de masa y de la cabeza, situada en las inmediaciones de la superficie magnetizable.

Una característica fundamental de la utilización de las memorias masivas es la forma de acceso, o sea el método empleado para tomar una información (lectura) o para grabarla (escritura).

La modalidad de acceso puede ser **secuencial** o **directa**.

Se tiene el acceso secuencial cuando, para obtener una determinada información, es necesario recorrer todas las grabaciones efectuadas hasta encontrar la que interesa.

Se tiene el acceso directo cuando es posible localizar inmediatamente la información que interesa conociendo su dirección física (número del disco, pista, sector).

Estos conceptos se expondrán más ampliamente en la parte correspondiente a los archivos.

Cintas magnéticas

Las cintas magnéticas se utilizan ampliamente para memorizar ingentes cantidades de datos, para la copia de archivos importantes (especialmente como medida preventiva contra la pérdida de datos) y para una conveniente elaboración de informaciones organizadas en archivos secuenciales.

La cinta consiste en un soporte de plástico, normalmente de una anchura de media pulgada (unos 12,5 mm) y de longitud variable; una cara está recubierta de una fina película de material magnetizable (como en las cintas magnetofónicas corrientes).

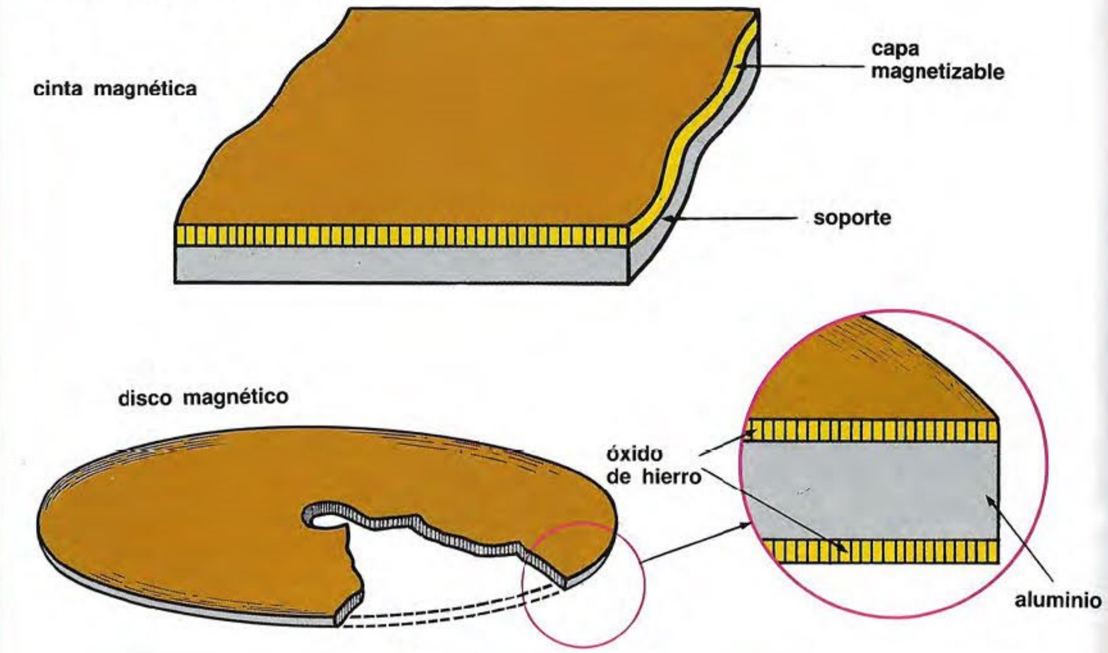
Dichas cintas magnéticas (llamadas también bobinas) pueden ser grabadas (escritura) o



Italcable

Detalles de unidades de cinta en un moderno centro de cálculo.

SOPORTES MAGNETICOS





Sarin

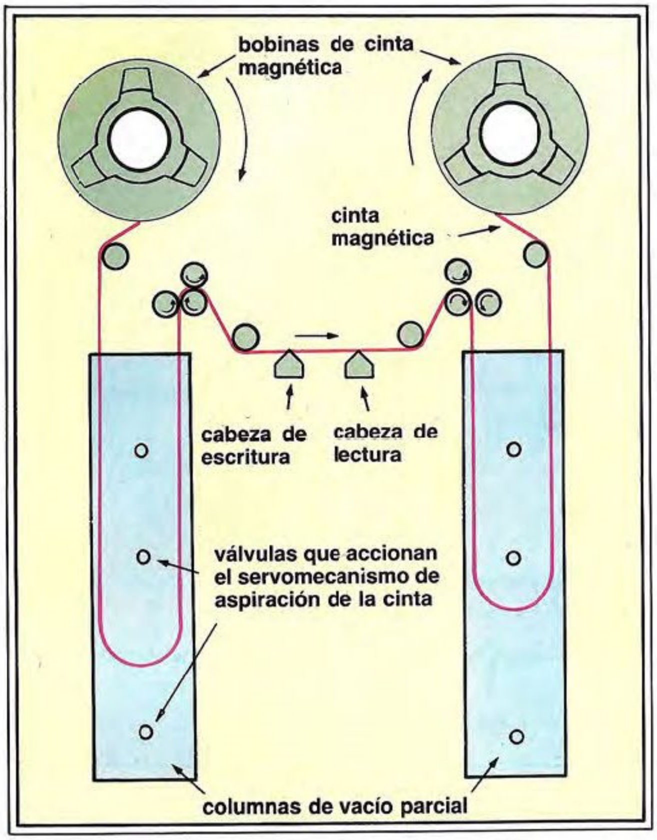
Unidades de memoria masiva de un banco de datos.

consultadas (lectura) mediante los accesorios adecuados, denominados **unidades de cinta** y que están considerados como unidades periféricas del ordenador.

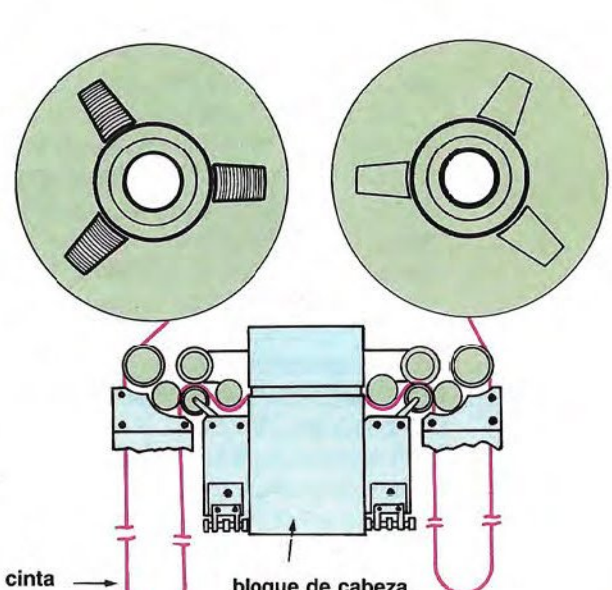
El dispositivo más importante de las unidades de cinta es el que se encarga de la lectura/escritura, así como de borrar la cinta magnética. La grabación se efectúa mediante la magnetización de puntos que pueden adoptar los dos estados asociados a los valores «1» (punto magnetizado) y «0» (punto no magnetizado).

La superficie de una cinta magnética se puede considerar subdividida en pistas (o canales) a lo largo de las cuales se magnetizan los puntos «0» y «1» (hay cintas de 7 y 9 pistas). La última pista se utiliza como «pista de paridad», y se usa para controlar la exactitud de las grabaciones en las otras 8 pistas (cintas de 9 pistas): el punto (bit) en ella memorizado adopta el valor «1» o «0» según que el número total de bits «1» contenidos en el carácter considerado sea par o impar (paridad transversal). Existe un control análogo para todos los bits de una pista (paridad longitudinal)*.

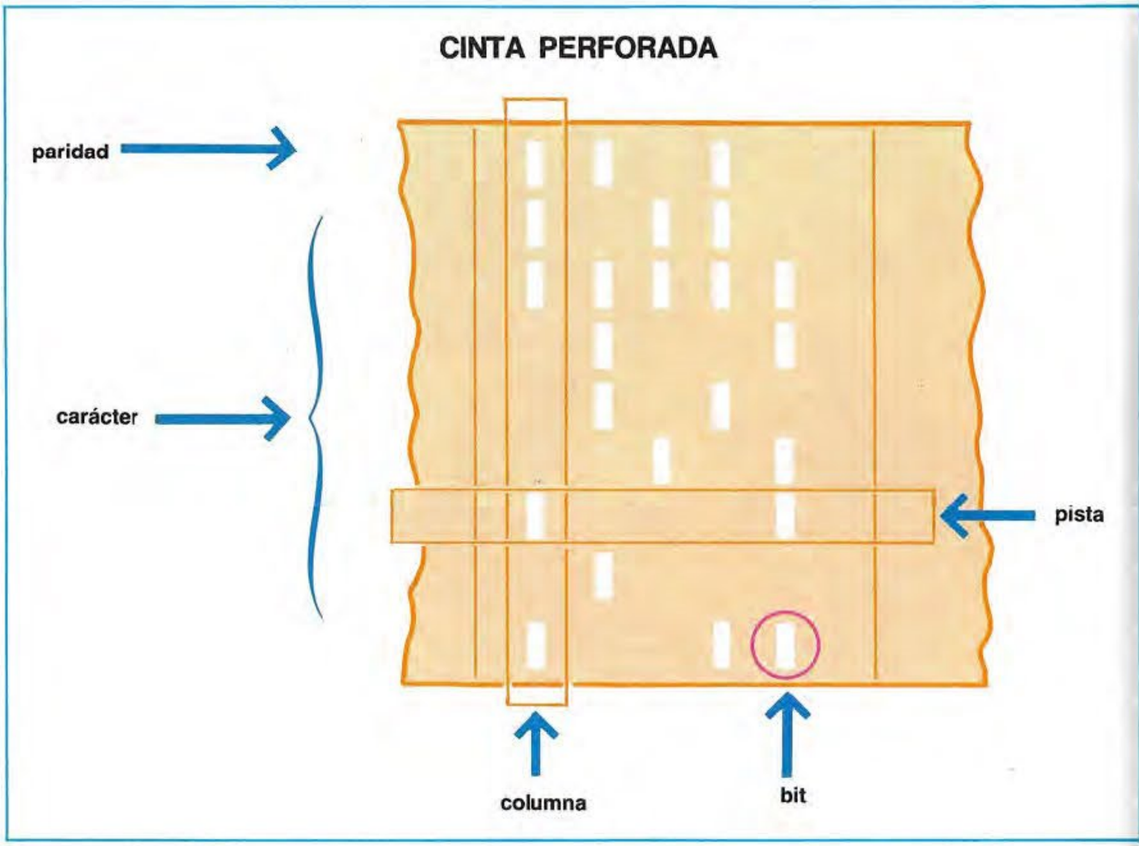
* Este método es del todo análogo al usado en el control de transmisión en código ASCII (bit de paridad, ver págs. 117 a 119).



ESQUEMA DE UNA UNIDAD DE CINTA



ESTRUCTURA DE UNA UNIDAD DE CINTA



La cantidad de caracteres que puede contener una cinta magnética depende de la longitud física de la misma y del grado de concentración con que se pueden grabar los caracteres (**densidad de grabación** o **empaquetado**).

En función de la longitud de la cinta magnética, se utilizan bobinas de varias medidas (600, 1.200, 2.400 y 3.600 pies).

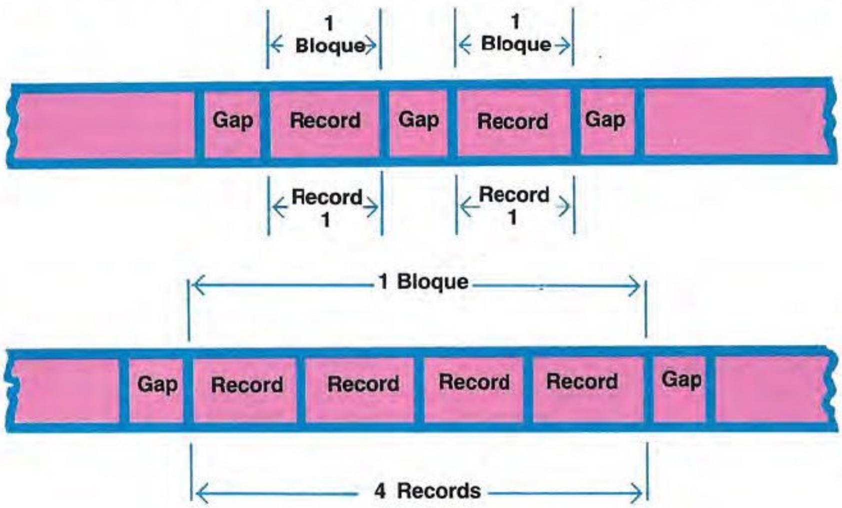
La densidad de grabación también puede variar, y los valores más comunes son: 256, 512, 800, 1.600 y 6.250 caracteres por pulgada.

Sobre la base de estos valores se puede determinar la capacidad teórica de una cinta magnética; por ejemplo, una bobina de 2.400 pies (unos 730 metros), grabada con una densidad de 800 caracteres por pulgada, tiene una capacidad teórica de aproximadamente 23 millones de caracteres.

En realidad, la grabación de caracteres en cinta magnética no es continua, puesto que los caracteres se distribuyen en paquetes o bloques, y cada bloque está separado del contiguo por un trocito de cinta sin grabar (**gap** o **interbloque**). La amplitud de los bloques varía normalmente en función de la organización de los da-

tos y de la capacidad de la memoria central del ordenador. La porción de cinta no grabada (gap) varía según el tipo de unidad de cinta, y es bastante pequeña (por ejemplo, 0,60 o 1,12 pulgadas).
Resulta, pues, evidente que la capacidad efecti-

ESQUEMA DE GRABACION DE DATOS EN CINTA MAGNETICA



va de una cinta magnética depende también de la longitud de los bloques y de la frecuencia de los gaps: hay mayor capacidad con bloques de muchos caracteres (menor número de gaps); hay menor capacidad con bloques de pocos caracteres (mayor número de gaps). Como ya se ha dicho, las unidades de cinta se conectan al ordenador como unidades periféricas.
El diálogo entre ambos aparatos tiene lugar mediante la transferencia de caracteres por cable (canal) a alta velocidad; la velocidad de transferencia de los datos puede tomar valores del orden de 20.000 a 60.000 caracteres por segundo, y llegar incluso a 120.000-320.000 caracteres por segundo.
Para garantizar una seguridad de uso adecuada, la cinta magnética ha de tener una cierta cantidad de cinta no grabable al comienzo (unos 3 metros) y al final (unos 4 metros) de la bobina, para permitir el deslizamiento a través del mecanismo de alimentación. El comienzo de la parte útil se indica con una chapita de aluminio de unos 25 mm (**indicador de comienzo de cinta**); de la misma forma se señala el final de la parte grabable (**indicador de final de cinta**). Di-



Rhône-Poulenc Italia

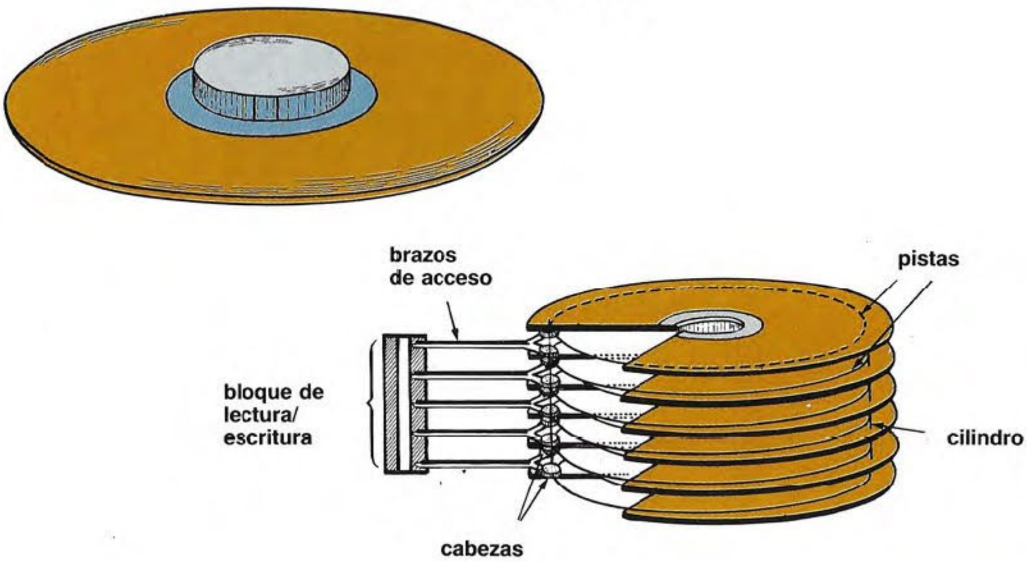
El equilibrado en dos planos confiere perfecta estabilidad a la fase de rotación de los discos en la unidad de disco.



Rhône-Poulenc Italia

La verificación digital en unidad periférica garantiza la adaptación del paquete de discos al ordenador.

PAQUETE DE DISCOS



chas chapitas permiten a las células fotoeléctricas de la unidad de cinta determinar dónde puede iniciarse (punto de carga) y dónde debe terminar la operación de lectura/escritura.

Discos magnéticos

Los discos magnéticos (unidades de memoria masiva de acceso directo) también se utilizan para contener grandes cantidades de datos. La característica que los distingue de las cintas magnéticas es la velocidad con que se puede obtener la información deseada. El disco magnético está constituido por un fino disco metálico con las caras recubiertas de material magnetizable (véase dibujo pág. 226).

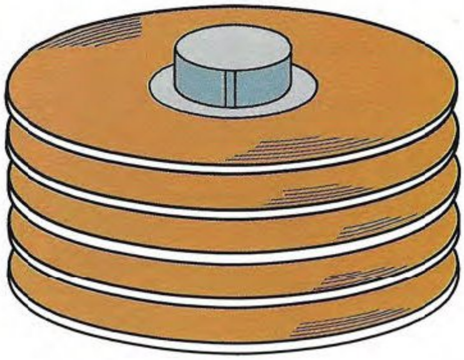
Varios discos pueden estar montados sobre un eje central y girar con velocidad uniforme. En ese caso, los diversos discos están separados entre sí lo suficiente como para permitir el movimiento de las cabezas de lectura/escritura.

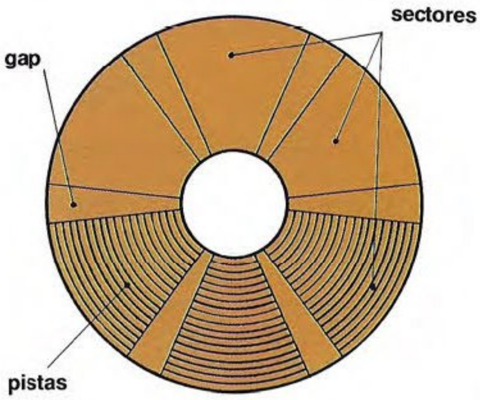
Las pistas son accesibles en lectura y escritura mediante el movimiento radial de las cabezas entre los discos en rotación.

Tal conjunto de discos se denomina **disk pack** (**paquete de discos**).

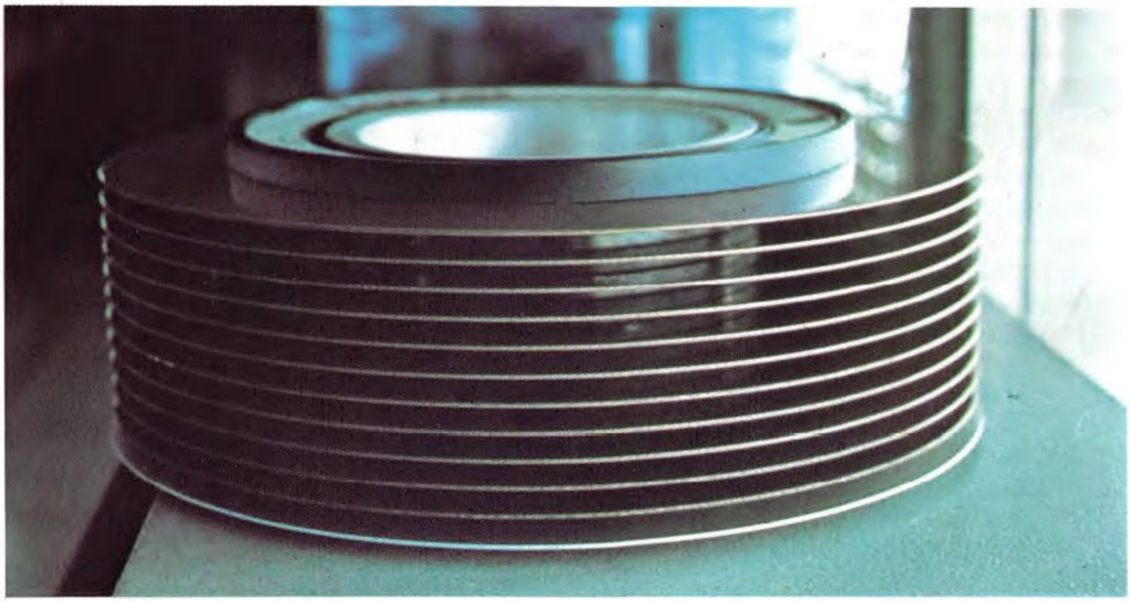
Los datos se graban en las pistas concéntricas según los mismos principios físicos utilizados en las cintas magnéticas. Las pistas (varios centenares en cada cara del disco) se subdividen en sectores, separados entre sí por intervalos sin grabar denominados gaps.

En las unidades de varios discos, las cabezas correspondientes a la cara inferior de un disco y



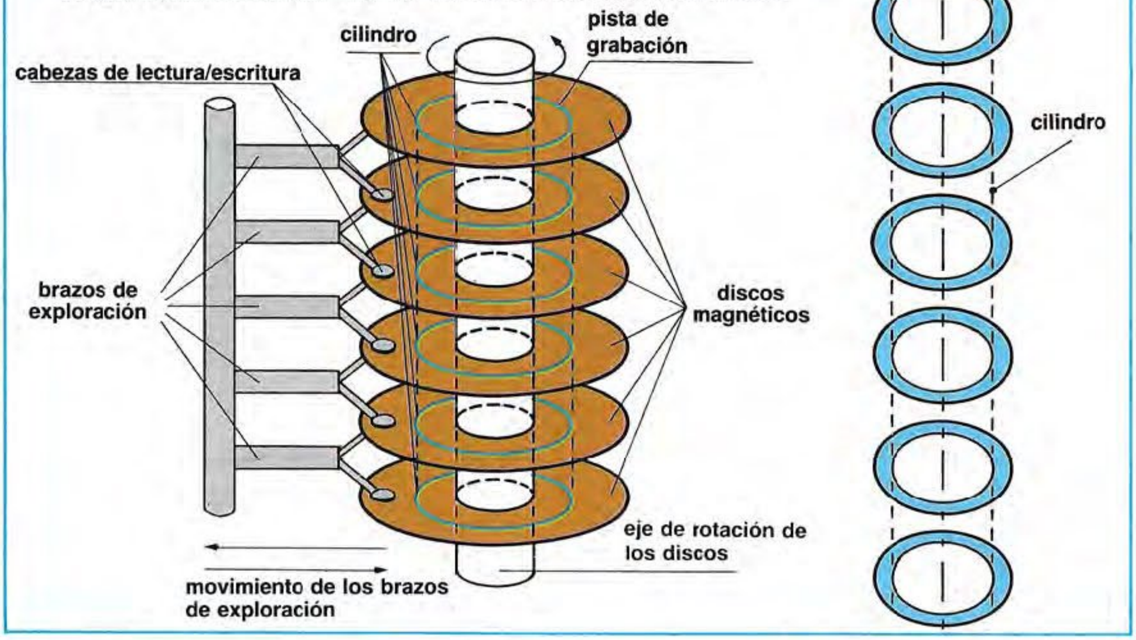


Paquetes de discos. Constituyen las memorias de masa de los sistemas de más envergadura.



Interna

ESQUEMA DE UNA "PILA" DE DISCOS MAGNETICOS



a la superior del disco de abajo están montadas en el mismo brazo mecánico.

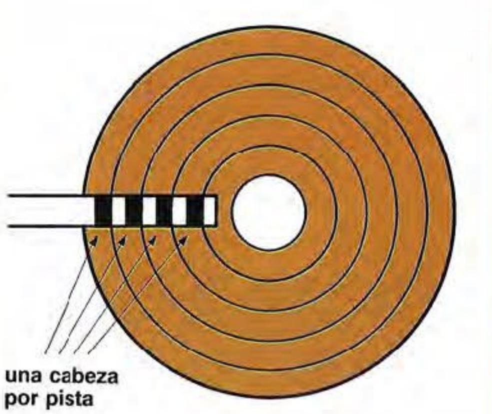
Puesto que el brazo se mueve solidariamente con todas las cabezas, éstas se hallan en la misma vertical y posicionadas sobre la misma pista de cada disco. La serie de pistas en una misma vertical se denomina **cilindro**.

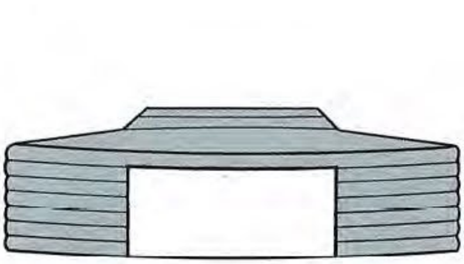
Las pistas pertenecientes a un mismo cilindro pueden ser, pues, leídas o escritas consecutivamente sin desplazar las cabezas, obteniendo un notable ahorro de tiempo.

La reducción de los tiempos de acceso se puede conseguir también mediante el empleo de cabezas fijas; en este caso se elimina el movimiento mecánico del brazo de exploración.

Debido al movimiento de rotación, el contenido de un sector sólo puede ser leído o escrito cuando dicho sector pasa bajo la cabeza de lectura/escritura. El tiempo de acceso a los datos es, por tanto, variable y depende de la suma de los siguientes factores (en el caso de cabezas móviles):

- tiempo de posicionamiento de la cabeza sobre la pista que contiene el sector buscado;
- tiempo de espera necesario para que el sector buscado pase bajo la cabeza;
- tiempo de lectura o escritura propiamente dicho.





Los tiempos globales son extremadamente bajos y pueden oscilar entre una decena y un centenar de milisegundos.
Las capacidades por módulo varían entre varios millones hasta 500-600 millones de caracteres.
El intercambio de datos entre las unidades de disco y el ordenador se efectúa con velocidades de transferencia que van desde varias decenas de millares hasta 1.800.000-2.000.000 de caracteres por segundo.
Los discos suelen estar encerrados en contenedores de plástico con las correspondientes aberturas para los brazos de exploración.

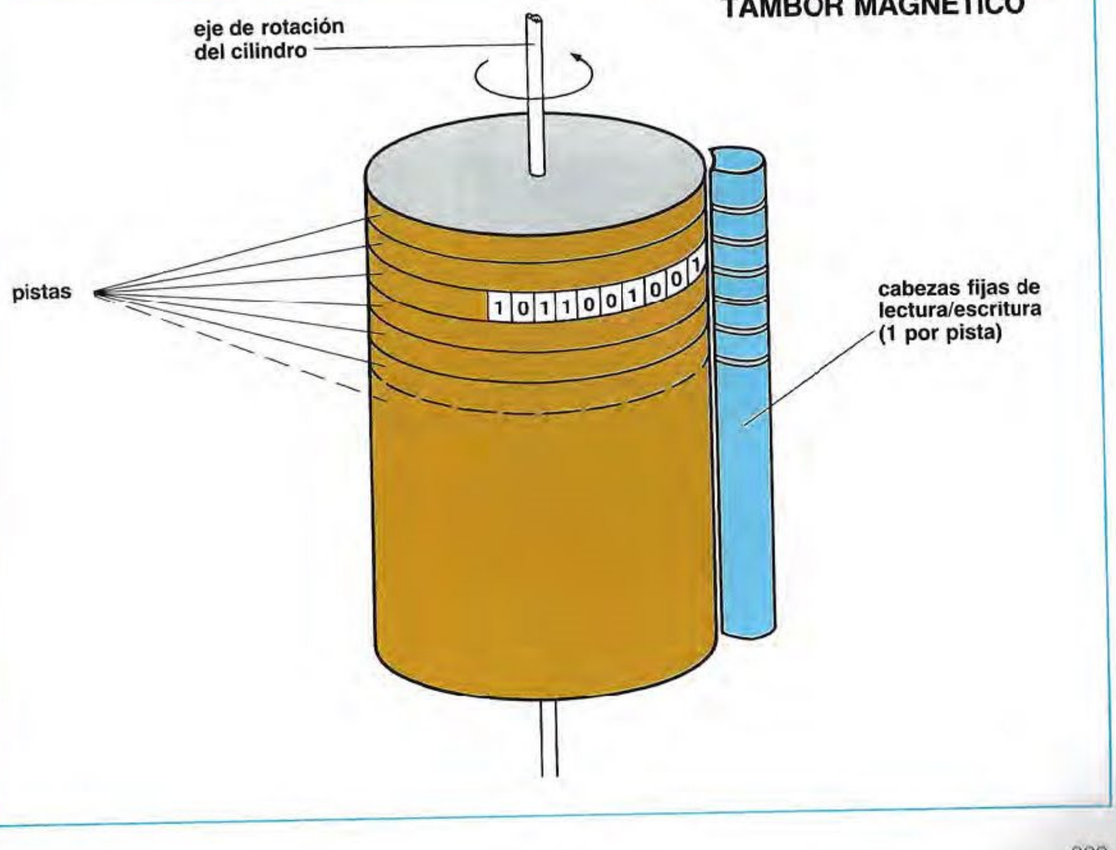
Tambores magnéticos

Los tambores magnéticos se utilizan, normalmente, en los grandes ordenadores, debido a su elevado costo.
Estas memorias masivas de acceso directo están constituidas por un cilindro de metal con la superficie lateral recubierta por una fina capa de sustancia ferromagnética, y giran a gran velocidad con movimiento uniforme.
La cabeza de lectura y grabación está posicionada en correspondencia con cada pista (que, a su vez, puede estar subdividida en sectores), a una distancia de algunas centésimas de milímetro.
En general, las cabezas son fijas y hay tantas como pistas.
La capacidad de un tambor magnético varía en función de su tamaño, del número de pistas, así como de la densidad de grabación de los datos en cada pista.
El tiempo de acceso a una información viene dado por la suma del tiempo necesario para esperar el paso del sector que contiene dicha información más el tiempo de lectura propiamente dicho. Para disminuir el tiempo de acceso se han fabricado tambores magnéticos con varias cabezas por pista.

Floppy

Los floppy son dispositivos más sencillos de manejar y de costo inferior que los discos, a los que conceptualmente son similares.
Estos discos son de material plástico flexible y van dentro de una funda provista de una ventanilla que permite el acceso de las cabezas de lectura/escritura.

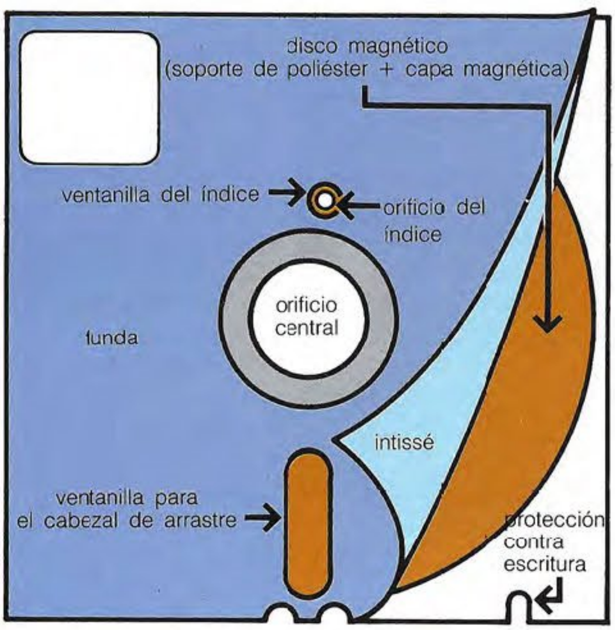
TAMBOR MAGNETICO



Centro THC



Floppy comparado con un disco normal de 45 r.p.m. En el floppy caben alrededor de 1.200.000 caracteres.



ESTRUCTURA DE UN FLOPPY FLEXETTE DE RHÔNE-POULENC

La funda
Está hecha a base de un PVC de excepcional resistencia a la temperatura.

El intissé (no-tejido)
La forma de enganche específica del intissé con que está formada la funda, permite al disco girar a velocidad constante y asegura que quede perfectamente centrado al cargarlo en el lector.

El disco
Se hace con un sustrato de poliéster recubierto de una emulsión magnética de alto poder reticulante.

El pequeño orificio redondo sirve para posicionar el disco en su alojamiento, a fin de obtener la alineación de las cabezas con el punto inicial de las pistas.
También los floppy están divididos en pistas, y cada pista subdividida en sectores.
La capacidad de un floppy varía en función de los siguientes elementos:

1 / grabación en una sola cara o ambas
2 / grabación con densidad sencilla o doble

Considerando estos elementos, la capacidad de un floppy varía entre 170.000 y 1.200.000 bytes aproximadamente. Además, puede haber varios floppy en un mismo cargador.
Normalmente, estos soportes se utilizan en los miniordenadores y ordenadores personales; los más comunes son los de 5 1/4 y 8 pulgadas.

Los ficheros

Los conjuntos de datos grabados en memorias masivas se denominan ficheros y se conocen también con el término inglés **files**.
Una característica de un fichero es la de contener datos homogéneos, como, por ejemplo, los nombres de los empleados de una empresa, los alumnos de una escuela o los productos de un almacén.
Examinemos ahora la estructura de estos datos.
El dato es un conjunto de uno o más caracteres que individualiza de modo preciso y unívoco cualquier elemento, como por ejemplo:

- **un nombre:** Antonio Valera
- **un número:** 358
- **unas siglas:** A2

Un dato puede ser de distintos tipos:

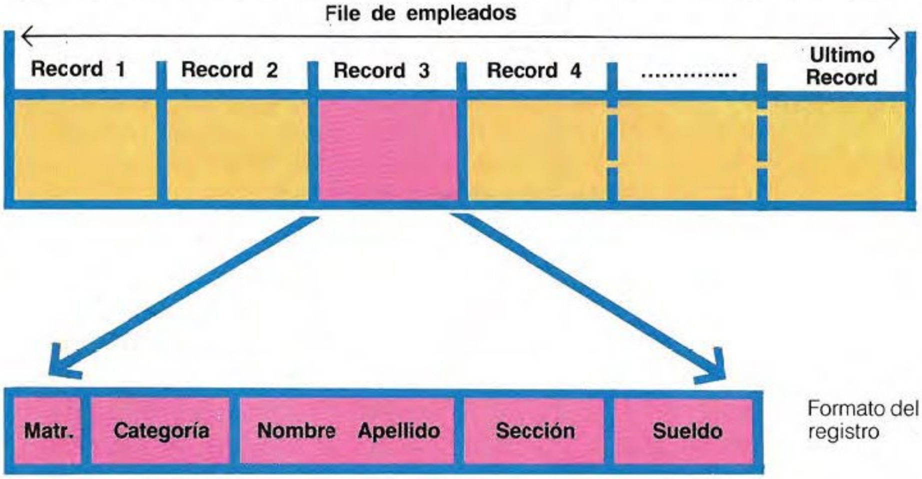
- **alfabético:** cuando sólo se compone de letras y/o espacios
- **numérico:** cuando sólo se compone de cifras
- **alfanumérico:** cuando se compone de caracteres cualesquiera (letras, cifras y espacios)

La porción de memoria (central o auxiliar) ocupada por un dato se llama **campo** (**field**), al que normalmente se asocia un nombre.

nominativo 40 car.	
apellido 25 car.	**nombre 15 car.**

En el ejemplo vemos que el campo **nominativo** está dividido en dos elementos (**subcampos**) y que a cada uno de éstos se da una dimensión concreta: 25 caracteres para el subcampo apellido y 15 caracteres para el subcampo nombre. Si consideramos el nominativo Antonio Valera, vemos que el apellido Valera utiliza 6 de los 25 caracteres disponibles, y que el nombre Antonio utiliza 7 de los 15 caracteres del otro subcampo; queda, por tanto, espacio no utilizado en cada subcampo. Es evidente que cada subcampo se ha previsto lo bastante amplio para poder contener los apellidos y nombres más largos del conjunto de ellos que debe ser examinado.

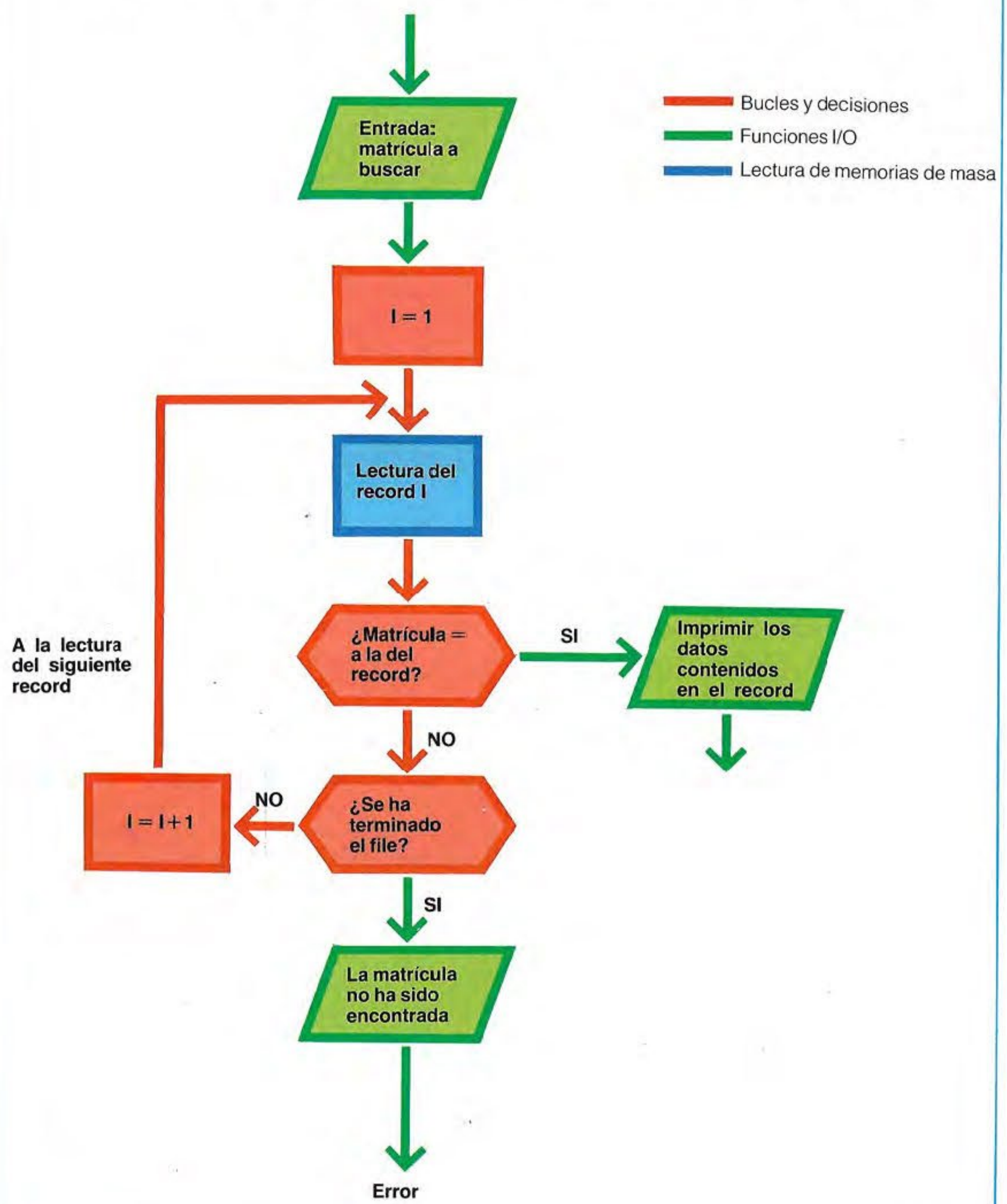
EJEMPLO DE ESTRUCTURA DE DATOS EN UN FILE DE EMPLEADOS



Formato del registro

Cuando el campo mantiene inalterada su amplitud, se denomina **de longitud fija**.
Cuando, por el contrario, la cantidad de memoria utilizada corresponde estrictamente al número de caracteres que constituyen el dato, el campo se denomina **de longitud variable**.
Leopoldo Altarriba, 18 caracteres (comprendido el espacio); María Benítez, 13 caracteres (comprendido el espacio); Antonio Valera, 14 caracteres (comprendido el espacio). El conjunto de

DIAGRAMA DE FLUJO DE UNA BUSQUEDA MEDIANTE "CLAVE"





Las grabadoras magnéticas destinadas a los ordenadores están entre las más sofisticadas.

los campos que califican el objeto de la grabación (dato) se denomina **record** o **registro**. Supongamos que queremos crear un archivo con los principales datos de los empleados de una empresa. Los datos a memorizar son:

1 / Número de matrícula (3 caracteres)
2 / Categoría (2 caracteres)
3 / Nombre y apellido (40 caracteres)
4 / Sección (2 caracteres)
5 / Sueldo (7 caracteres)

El conjunto de estos datos (1,... 5) constituye un record, y el conjunto de todos los records es el fichero. En él, cada record está formado por cinco campos, cada uno de los cuales contiene uno de los cinco datos. El número total de records es igual al número de empleados. En la pág. 231 se muestra la estructura de este file.

Al objeto de individualizar de forma precisa y unívoca todos los datos presentes en un record, normalmente uno o más campos juegan el papel de **clave**; en nuestro ejemplo, los empleados de la empresa son identificados por el número de matrícula, y cada número de matrícula se convierte en una clave mediante la cual se puede tener acceso a los datos. En otras palabras, para obtener los datos de un empleado habrá que introducir el número de matrícula: la máquina iniciará la lectura del fichero a partir del primer record comparando la matrícula suministrada por el operador con las de los datos, y seguirá, un record tras otro, hasta encontrar el valor coincidente. Al producirse esta coincidencia presentará los datos contenidos en el record y terminará la tarea. En la pág. 232 se puede ver el diagrama de flujo de este sencillo programa. Este método de búsqueda (lectura de todos los records de forma secuencial) puede utilizarse sólo cuando los datos no son muy numerosos. Para los grandes archivos hay métodos mucho más rápidos pero también más complejos.

Igual que ocurre con los campos de un record, el propio record puede ser:

- **record de longitud fija:** si todos sus campos son de longitud fija
- **record de longitud variable:** si contiene al menos un campo de longitud variable, o un número variable de campos

Los records que acabamos de ver se denominan **records lógicos**, y son elementos de un fichero que contiene uno o más datos relativos a

El hombre que inventó los videojuegos

¿Le dicen algo los nombres de Bell, Edison o Fermi? Pues bien, añada el de Higinbotham, el hombre que un día, hacia 1958, hizo aparecer una determinada imagen en su osciloscopio. Porque Willy, aquel día, había inventado el videojuego. He de decir que antes de llegar a esta conclusión he consultado una gran cantidad de manuscritos para poner orden en la historia de los videojuegos. Así que si sabe de un videojuego anterior a 1958 (y puede aportar pruebas), dígalo; de lo contrario, deje que Willy Higinbotham ocupe el puesto que le corresponde en la historia. Los videojuegos los ha inventado él y lo ha demostrado, les guste o no a las grandes empresas que se disputan su paternidad.

En los años cincuenta, las visitas organizadas a los laboratorios eran bastante aburridas: se enseñaban fotografías y se ilustraba uno u otro aspecto de una investigación. Willy, que para entonces había descubierto su inclinación por la física en Cornell y por la electrónica en el MIT, decidió hacerlas un poco más animadas: «¿No podríamos hacer participar a los visitantes en algún juego en una pantalla, de forma que manejen instrumentos y se diviertan apretando botones?» Dicho y hecho: Willy y sus colegas inventaron un partido de tenis que se jugaba en la pantalla de cinco pulgadas de un osciloscopio.

En 1958 ya habían aparecido los primeros ordenadores digitales, y precisamente la sección en la que trabajaba Willy había fabricado uno. Para su juego, utilizó un ordenador analógico, que representaba los datos mediante valores de tensión en lugar de impulsos on/off, al que había conectado un conjunto no programable de relés electromecánicos, potenciómetros, resistores, condensadores y amplificadores operacionales («op-amp», para los especialistas). El propio Willy admite que el armatoste no era especialmente elegante. Pero funcionaba, aunque había sido construido en poquísimo tiempo con componentes disponibles en el momento, incluidos algunos flip-flop de invención reciente que abrían el camino a nuevos desarrollos.

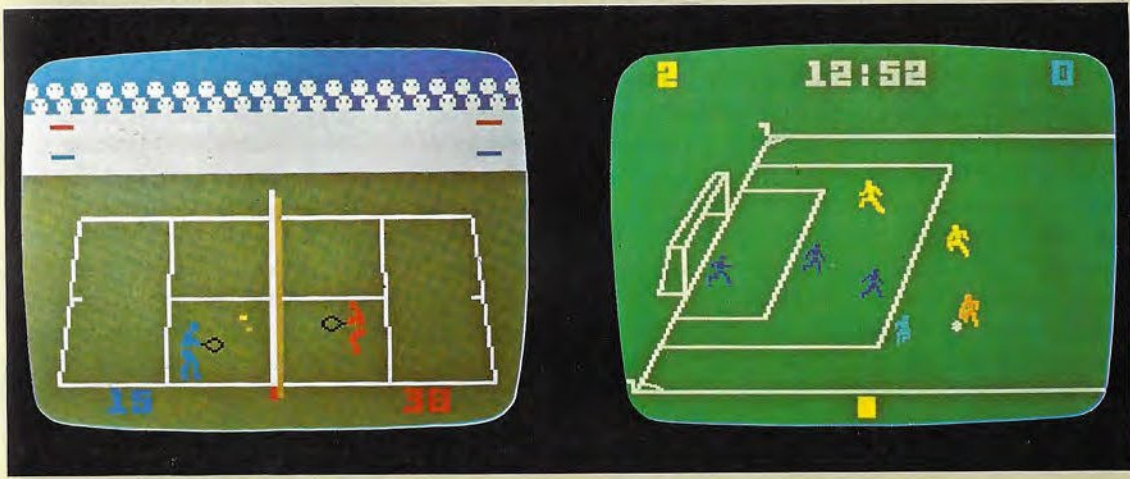
En la pantalla aparecía la representación de un campo de tenis visto de lado: una «T» invertida, con el palo vertical más corto, que representaba la «red». Cada jugador sujetaba una cajita provista de una palanquita y un pulsador. La palanquita servía para controlar el ángulo de tiro, y el pulsador se utilizaba para elegir el momento en que la pelota era lanzada (un momento cualquiera, con tal de que la pelota estuviera en el propio lado de la red). El sistema tenía en cuenta la gravedad, la velocidad del viento e, incluso, los rebotes; por ejemplo, si la pelota tocaba la red, el rebote era más débil que si chocaba directamente contra el suelo.

El juego era sencillo, pero divertido y apasionante. Algunos colegas de Higinbotham aún recuerdan que los estudiantes estaban entusiasmados y que «no había manera de hacer que se fueran». Puedo imaginarme perfectamente a Dave, mi jefe: un jovenzuelo pegado a la pantalla, con una misteriosa vocecita en su interior diciéndole: «Recuérdalo, puede ser importante».

La pelota y las líneas del campo eran redibujadas continuamente en secuencia, a una velocidad que permitía una visión sin fluctuaciones: se trata de una solución todavía en uso. Pero lo que era y ha permanecido único es la forma en que se hacía mover la pelota. Intentemos dar una idea sin adentrarnos demasiado en cuestiones técnicas.

Un osciloscopio puede servir para visualizar gráficos de tipo dinámico en los que las coordenadas del punto luminoso son proporcionales a los valores de las tensiones aplicadas a las entradas x e y. Higinbotham pensó en utilizar amplificadores operacionales para realizar un circuito que simulaba la trayectoria de una pelota y el impacto en el suelo. En el momento del choque se excitaba un relé que invertía la polaridad de otro amplificador. De esta manera se hacía que la pelota volviera atrás y daba la impresión de rebotar. Primitivo, pero eficaz. Con el mismo método se podía establecer si la pelota había tocado la red, en cuyo caso se hacía rebotar con una velocidad menor para dar al juego mayor realismo. La velocidad de la pelota, continuamente frenada por la velocidad del viento, se simulaba directamente con un resistor de 10 megaohmios.

Los jugadores tenían una palanquita para escoger el lado del saque, y también podían regular la altura de la red y la longitud del campo. No había peligro de que un jugador no le diera a la pelota: pulsando el botón se la «golpeaba» siempre, con tal de que estuviera en el lado correcto de la red; pero, si no se escogía el momento y el ángulo adecuados de devolución, la pelota no volvía al campo contrario.



Marka

La instalación era mucho más sofisticada que los primeros juegos de ping-pong electrónico, y durante dos años fue el plato fuerte de los Laboratorios Brookhaven. Pero en un momento dado el aparato fue desmantelado.

Le he preguntado a Willy cómo es que no patentó este descubrimiento, teniendo en cuenta que ostentaba ya más de veinte patentes, todas ellas transferidas al gobierno de los Estados Unidos. «Era divertido, y nos dábamos cuenta de que tenía posibilidades, pero no era algo que pudiera interesar al gobierno. Por otra parte, tal vez haya sido mejor así. De lo contrario, hoy, para proyectar un videojuego, haría falta una licencia gubernativa.» Ríe: la idea parece divertirle.

La que no ha reído ha sido Magnavox: los derechos sobre los videojuegos pueden significar millones de dólares, y son algo muy, muy serio. Esta empresa ha pedido una patente para los videojuegos basados en pelotas que rebotan, y ha recibido de Higinbotham declaraciones juradas sobre su invento. No es que Willy piense obtener nada en términos de dinero, pero pretende defender su reputación y su prestigio.

Pero ¿quién es realmente este hombre? ¿Qué significado ha tenido en su vida ese campo de tenis en la pantalla de un osciloscopio? Para intentar comprenderlo hemos de repasar las principales etapas de su carrera.

Al estallar la II Guerra Mundial, Higinbotham acababa de graduarse en física en la Cornell University. Le ofrecieron tomar parte en las investigaciones del MIT sobre radiaciones, y trabajó en la puesta a punto de una importante e innovadora técnica entonces conocida como «Radio Detecting and Ranging». ¿No le dice gran cosa? Piense que más tarde se abrevió como RADAR.

Seguidamente, Willy participó en el Manhattan District Project y trabajó como físico en otra tecnología exótica y potencialmente importante. En 1945, convertido en jefe de la Sección Electrónica, proyectó los circuitos temporizadores que contaron los últimos milisegundos anteriores a la explosión de la primera bomba atómica.

En el momento de la explosión en Los Alamos, Willy se hallaba a 40 kilómetros de distancia, y pudo observar todas sus fases a través de unas gafas de soldador con cristales tan gruesos que no podía ver un faro encendido ante sus narices. Le pido que me cuente algo de aquella experiencia, pero Willy se queda silencioso y pensativo. Luego sólo me dice que él y los demás observadores subieron a los autocares e hicieron todo el largo trayecto de vuelta en el más completo silencio. «Nadie tenía nada que decir.»

Los dos años siguientes Willy los pasó en Washington como secretario ejecutivo de la Federación de Científicos Americanos y, en su posición de funcionario de conexión entre el Congreso y los científicos, trabajó activamente a favor de la no proliferación de armas nucleares.

Hoy, decano de los Laboratorios Brookhaven, Willy ha recopilado, junto con sus colegas, la más completa documentación sobre dispositivos de protección nuclear que hay en el mundo. ¡Y pensar que también ha inventado los videojuegos!

(John Anderson. Extracto de DATA MANAGER, abril 1983.

un objeto perteneciente al file.
Los dispositivos empleados para leer/escribir en la memoria de masa permiten grabar un determinado record utilizando porciones del soporte (cinta, disco) comprendidas entre dos gaps. El tramo de memoria de masa implicado en estas operaciones se denomina **record físico**, y su tamaño varía en función de la amplitud de la parte de la memoria central que está destinada a acoger los datos.
Al establecer los records físicos, nos hallamos ante la exigencia de reducir los tiempos de acceso a los datos aumentando todo lo que sea posible el número de datos alojados en la memoria masiva. Estas exigencias se han de compatibilizar con la disponibilidad de espacio en la memoria central.
Volviendo a los records de los empleados de la empresa en cuestión, podemos grabar los datos en el soporte magnético (por ej., cinta) de dos maneras distintas:

Ejemplo 1

gap	record B	gap	record A	gap

I ← Bloque → I

Ejemplo 2

gap	record C	record B	record A	gap

I ←——— Bloque ———→ I

(en este caso, el record físico es equivalente a 3 records lógicos)
En el ejemplo 1, siempre hay intervalos no grabados (gaps) entre dos records, mientras que en el ejemplo 2 sólo hay gaps entre dos paquetes de records (bloques).
Con esta segunda modalidad de grabación hemos reducido las porciones no grabadas, aumentando el número de records lógicos grabables en la cinta.
A la vez, hemos reducido el tiempo necesario para las operaciones físicas de lectura/escritura, puesto que la unidad de cinta pone a nuestra disposición tres records lógicos a la vez.
El conjunto de los records agrupados se denomina bloque o record físico, y **factor de bloqueo** es el número de records lógicos del bloque (tres en el ejemplo).
La elección del factor de bloqueo ha de tender a reducir al máximo el número de gaps, de forma compatible con la disponibilidad de la memoria central para acogerlos.
Conviene observar que, cuando los records son de longitud fija, normalmente se agrupan en bloques con el mismo número de records, y por lo tanto los bloques son de longitud fija. Sin embargo, en el caso de records lógicos de longitud variable, se suele definir para los bloques una longitud máxima, y en cada bloque se agrupa el mayor número posible de records evitando su fraccionamiento. Por tanto, los bloques tienen longitudes variables, y su tamaño es siempre inferior o igual al máximo establecido. Cuando en un bloque se graba un solo record, el record físico coincide exactamente con el record lógico.
En los microordenadores y ordenadores personales, la longitud de un record físico es, para casi todas las máquinas, de 128 bytes.
En el ejemplo anterior (fichero de empleados), con las longitudes de campo consignadas tendríamos un record lógico largo: $3 + 2 + 40 + + 2 + 7 = 54$. Por tanto, en un record físico pueden entrar dos records lógicos enteros ($128/54 = 2,...$).

Organización de archivos en cinta magnética

Un archivo memorizado sobre cinta magnética es una lista lineal de records lógicos, donde los datos grabados pueden pasar uno tras otro bajo las cabezas de lectura/escritura.
Una característica fundamental de estos archivos es, por tanto, su estructura secuencial, de la que deriva una modalidad de grabación específica.
La grabación de los records tiene lugar, como hemos visto, de forma secuencial y, por consiguiente, la búsqueda de los records se efectúa leyendo el archivo desde el principio hasta encontrar el record buscado.
Las grabaciones pueden efectuarse sin orden alguno o bien pueden organizarse según una secuencia ascendente o descendente. La ordenación de los records memorizados en desorden se efectúa normalmente con el adecuado programa de selección (**Sort**), que suele formar parte del repertorio de programas del sistema.

Volviendo al ejemplo de los empleados, los datos se pueden grabar de las formas siguientes:

1 / sin ningún orden de matrícula

matr. 15 Antonio Valera
matr. 2 Leopoldo Altarriba
matr. 7 Roberto Martínez
matr. 5 Antonio Blanco
matr. 20 Francisco Viola

....

2 / en orden de matrícula creciente

matr. 2 Leopoldo Altarriba
....
matr. 5 Antonio Blanco
....
matr. 7 Roberto Martínez
....
matr. 15 Antonio Valera
....
matr. 20 Francisco Viola
....

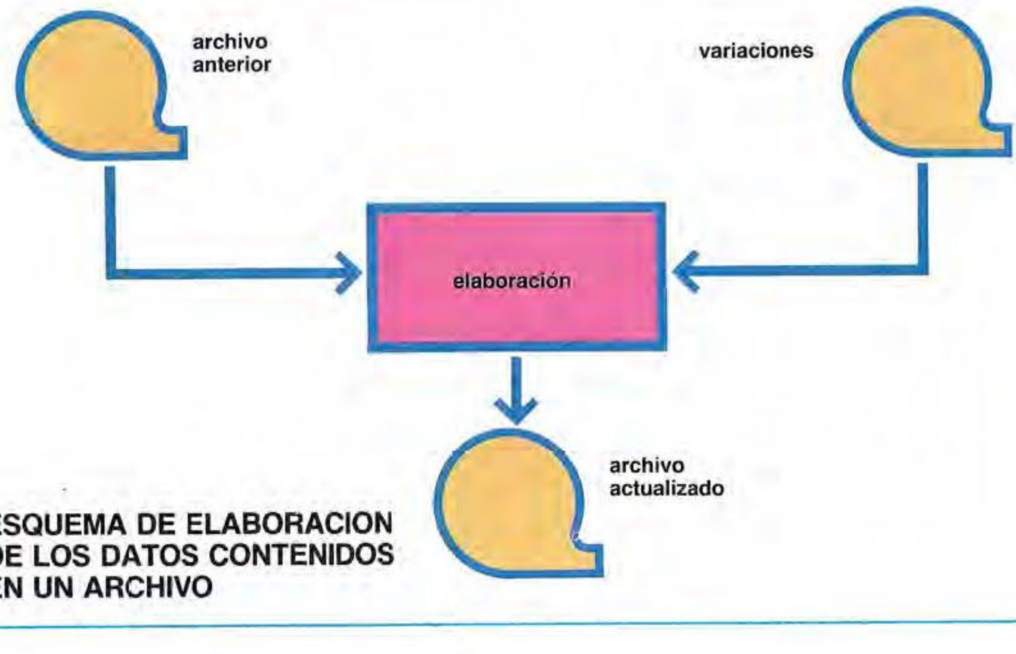
3 / en orden de matrícula decreciente

matr. 20 Francisco Viola
....
matr. 15 Antonio Valera
....
matr. 7 Roberto Martínez
....
matr. 5 Antonio Blanco
....
matr. 2 Leopoldo Altarriba

En el ejemplo, el orden por matrícula del empleado coincide con el orden alfabético. Esta doble ordenación se ha elegido sólo para facilitar la comprensión del ejemplo; en realidad la secuencia alfabética no cuenta, porque lo determinante es el orden de contratación (inscripción en la nómina).
Es evidente que se logra una mayor facilidad de búsqueda organizando el archivo en forma creciente o decreciente.
La organización secuencial más frecuente en los archivos en cinta magnética es en orden creciente.
La búsqueda se detiene en el momento en que se encuentra el record con el código requerido (hallado), o bien cuando el código del record examinado es mayor que el del requerido (no hallado).
Conocida la técnica de organización de los datos, es fácil el planteamiento de la puesta al día de un archivo.
Supongamos que tenemos que actualizar el archivo de la empresa, organizado de forma creciente, con las variaciones habidas en un determinado mes:

matr. **7** cambio de dirección
matr. **15** Antonio Valera despedido
matr. **21** María Viola nueva empleada

El esquema de la elaboración es el siguiente:



ESQUEMA DE ELABORACION DE LOS DATOS CONTENIDOS EN UN ARCHIVO

El archivo actualizado contendrá los siguientes records (si se adopta el criterio de la cancelación física):

matr. **2** Leopoldo Altarriba

....

matr. **5** Antonio Blanco

....

matr. **7** Roberto Martínez (con su nueva dirección)

....

matr. **20** Francisco Viola
matr. **21** María Viola

Organización de archivos en disco magnético

Mientras que en los archivos en cinta el acceso a una determinada información es necesariamente de tipo secuencial, las peculiares características físicas de los discos permiten acceder a los datos de distintas formas.

Efectivamente, los records pueden ser estructurados y conectados entre sí de forma tal que se facilite la búsqueda de los datos contenidos en los distintos campos y se reduzcan los tiempos necesarios para completarla.

Los tipos de organización más frecuentemente adoptados en los archivos en disco son cuatro:

- organización secuencial
- organización secuencial concatenada
- organización secuencial con índice
- organización con acceso directo

Organización secuencial

Los records se graban uno tras otro según el criterio de orden indicado en el campo clave del record. Este tipo de organización es muy idéntico al descrito para las cintas magnéticas, y el acceso al record deseado se logra leyendo antes todos los records que lo preceden.

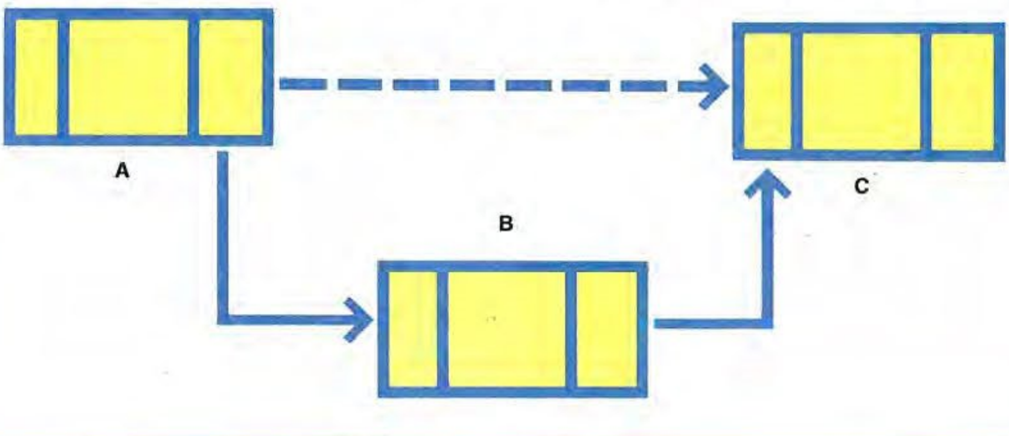
También en este caso se puede intervenir con un SORT para ordenar, según el campo clave, los records de un archivo realizado sin orden.

Sin embargo, son posibles algunas operaciones de actualización sin recurrir a la transcripción de un nuevo archivo. Podemos borrar un record situando un *marco* en un campo previsto a tal efecto, provocando, pues, una cancelación lógica en la secuencia de los records, sin efectuar una cancelación física (la información sigue presente, pero adecuadamente etiquetada). Podemos modificar el contenido de un record lógico, reescribiendo, empero, todo el bloque físico al que pertenece; podemos también insertar nuevos records pero sólo al final del archivo. Aunque ofrece las ventajas derivadas de la velocidad del soporte y de las posibilidades de intervención que acabamos de ver, la organización secuencial no aprovecha al máximo las posibilidades de los discos magnéticos.

Organización secuencial concatenada

Se dice que un conjunto de records está organizado en forma secuencial concatenada cuando en cada record, junto al dato propiamente dicho, se insertan algunas indicaciones que permiten localizar el record que contiene el dato siguiente. Introduciendo en el record elementos de referencia al record siguiente, podemos construir una cadena lógica y perfectamente ordenada de acuerdo con el campo clave.

INTRODUCCION DE UN DATO MEDIANTE LA MODIFICACION DE LOS PUNTEROS



Estas referencias a otro record se denominan **punteros (pointers)**. El record queda compuesto de esta manera:

clave	datos	puntero

Con esta técnica se facilitan las operaciones de actualización derivadas de la inserción de un nuevo record o de la cancelación de un record previo, puesto que basta con modificar la cadena de los punteros.

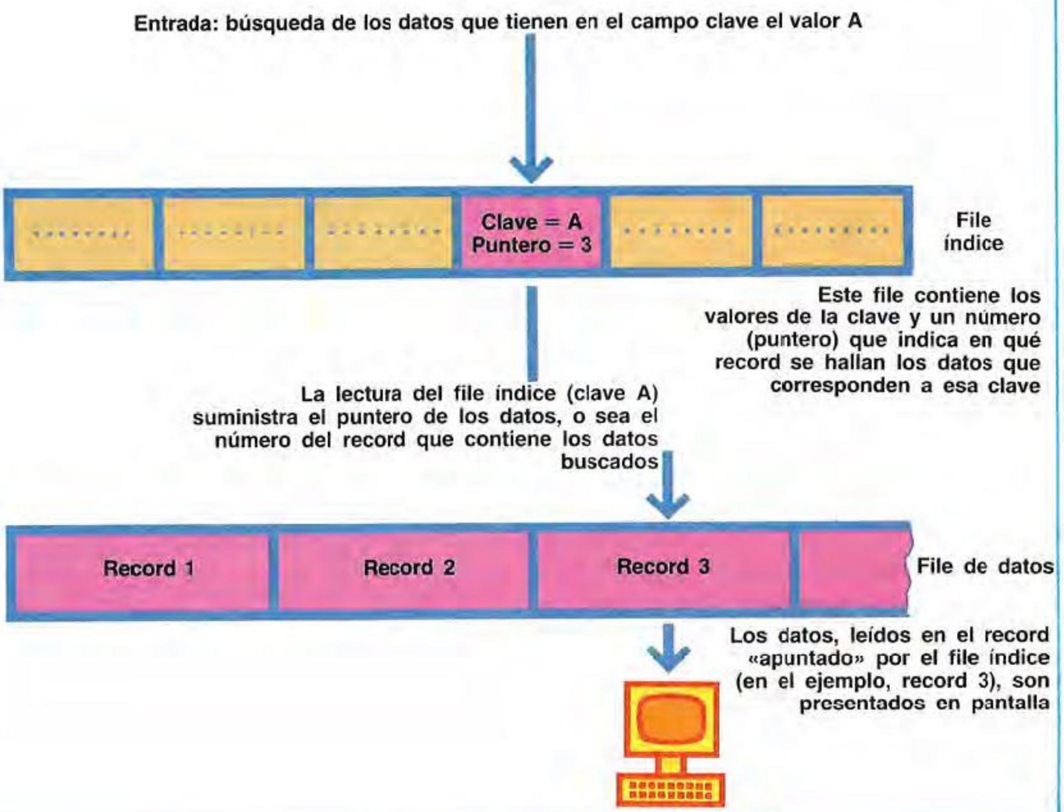
A / Ejemplo de inserción

Para insertar un nuevo record, B, entre los records A y C, se modifica el puntero de A para ligarlo a B y en éste se pone el puntero hacia C

B / Ejemplo de cancelación

Para cancelar el record B en la cadena de records ABC, basta con modificar el puntero de A haciendo que apunte a C

ESQUEMA BASICO DE BUSQUEDA DE DATOS EN UN FILE SECUENCIAL CON INDICE



De esta forma no se altera la secuencia física de grabación, y los nuevos records se insertan en otra zona del disco, denominada **zona de overflow (desbordamiento)**, que se limpia periódicamente efectuando la transcripción del archivo en forma secuencial.

Organización secuencial con índice

El elemento fundamental que permite introducir en un archivo este tipo de organización es el **campo clave**, que se inserta en cada record y contiene un valor (clave de acceso) que sirve para identificar dicho record de forma unívoca. El archivo se sigue gestionando como una organización de tipo secuencial y ha de ordenarse de acuerdo con la clave de acceso.

Sin embargo, se le asocia un segundo archivo que hace de índice.

El archivo secuencial con índice resulta, pues, compuesto por dos elementos:

- archivo índice
- archivo de datos

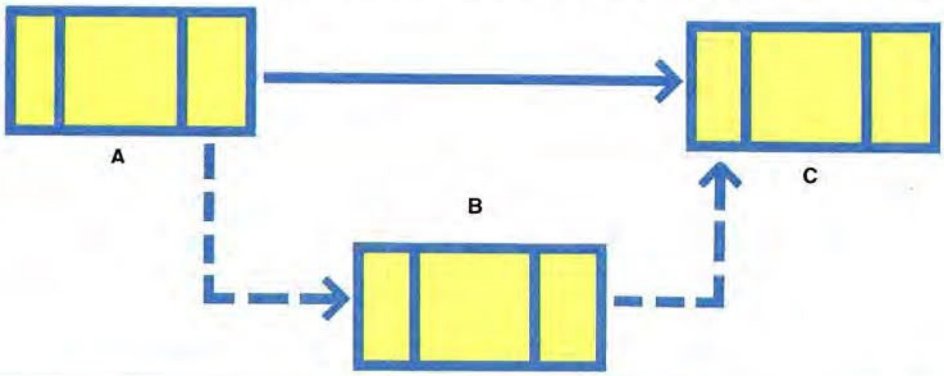
En el gráfico de pág. 239 vemos un esquema básico de este tipo de estructura.

La tabla registrada en disco como fichero secuencial constituye el índice del archivo.

A un mismo archivo pueden asociarse índices a distintos niveles, reduciendo aún más los tiempos de acceso.

Recordemos que el archivo de datos ha de es-

CANCELACION DE UN DATO MEDIANTE LA MODIFICACION DE LOS PUNTEROS



TEST 6

1 / ¿Cuáles de estas afirmaciones, relativas a la memoria de masa, son ciertas?
a: tiene una capacidad igual a la del ordenador;
b: las más veloces son las de cinta magnética;
c: permiten memorizar enormes cantidades de datos.

2 / ¿Cuántos caracteres pueden caber en un floppy? Suministrar los órdenes de magnitud superior e inferior.

3 / ¿Qué son: a: los files; b: los records; c: los campos?

4 / ¿Cuál es la utilización normal de las cintas magnéticas?

5 / Reescribir en orden jerárquico los nombres: byte, file, bit, record, archivo, campo.

Las soluciones, en las págs. 244 y 245.

tar rígidamente ordenado y que el valor del campo clave ha de ser tal que identifique biunívocamente al record correspondiente.
Para ilustrar mejor este tipo de organización, podemos considerar dos ejemplos típicos. Si el archivo de los productos de un almacén está grabado en un monodisco a partir del cilindro 3, podemos crear el siguiente archivo índice:

Archivo índice

Puntero al dato (n.° del cilindro)	Clave de acceso al dato (última clave para el cilindro)
3	1450
4	2972
5	4250
6	5587
7	6999

El acceso a un dato que tiene una determinada clave tiene lugar procediendo antes a la lectura secuencial del archivo índice, confrontando la clave de búsqueda con la tabla grabada en él. Esta operación suministrará el puntero al dato, en este caso el número del cilindro en el que

Operadora ante el teclado.



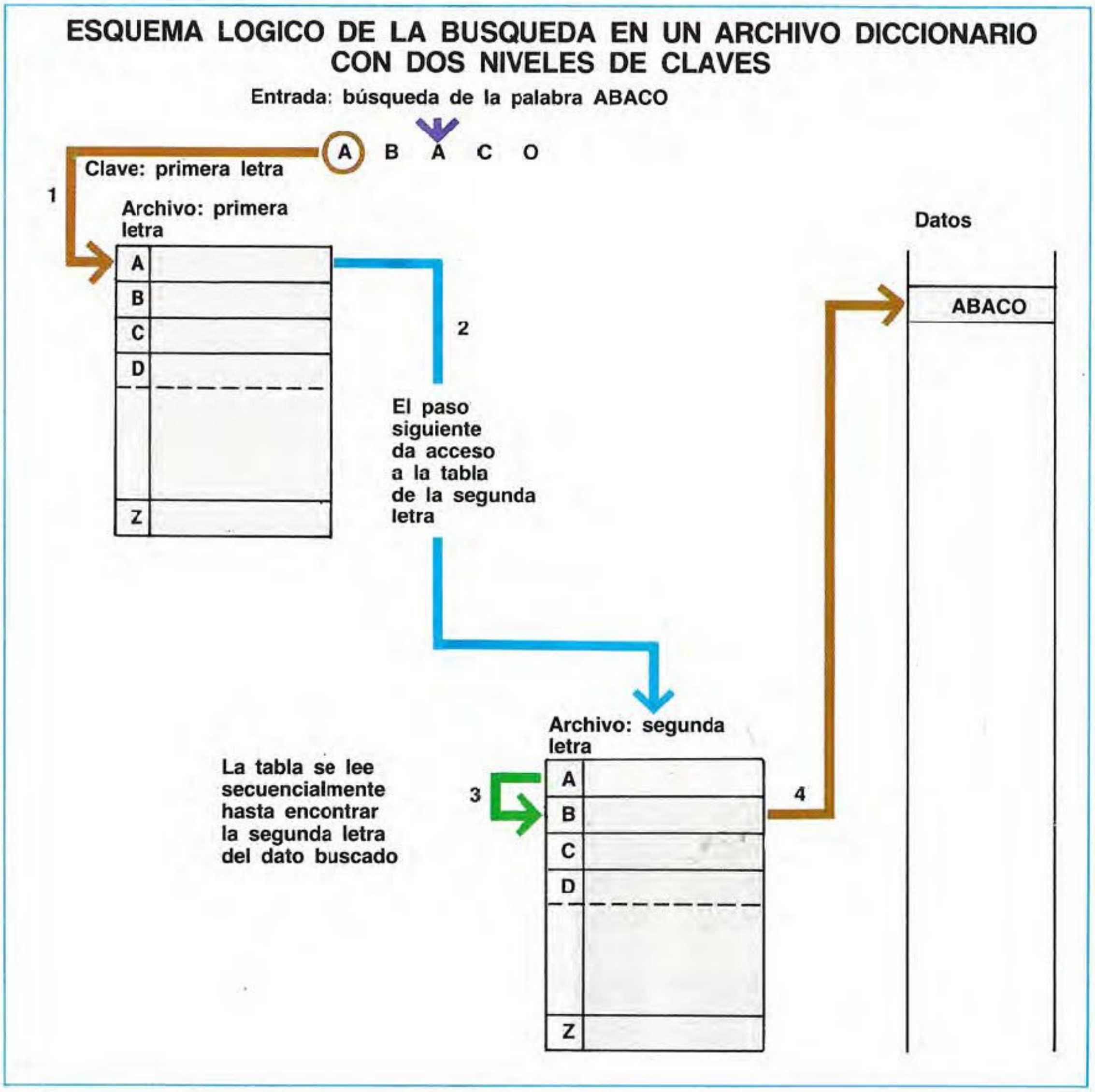
C. O'Rear/West Light-Grazia Neri

está grabado el record que contiene el dato buscado. La lectura del archivo de datos se limita, así, a los records grabados en el cilindro direccionado por el archivo índice.
Si se ha de buscar el producto cuya clave es 5350, la lectura del archivo de datos se limita sólo a los records grabados en el cilindro 6.
Como segundo ejemplo consideremos a gestión de un archivo cuyos elementos sean las denominaciones de un diccionario.
Si dividimos el archivo en subarchivos en base a las iniciales de las denominaciones, tendremos una tabla de 29 letras (índice de primer nivel). Cada grupo de denominaciones puede, a su vez, subdividirse en subarchivos considerando la segunda letra de cada palabra; establecemos así otras 29 tablas, compuestas también por las letras del alfabeto (índice de segundo nivel).
A cada elemento del índice de primer nivel le adjudicamos la dirección de la segunda tabla en la que están memorizadas las segundas letras de las denominaciones; concretamente, tenemos la siguiente relación:

A { AA, AB, ..., AZ } ..... { ....., ... }

B { BA, BB, ..., BZ } **Z** { ZA, ZB, ..., ZZ }

Direccionando con el segundo índice el archivo correspondiente, podemos proceder a su lectura y encontrar la palabra buscada.
Las operaciones a efectuar para la búsqueda de la palabra son las siguientes:
a) acceso al índice de primer nivel y lectura secuencial de los elementos contenidos en él hasta encontrar la letra igual a la inicial de la denominación buscada; el puntero direcciona al índice de segundo nivel;
b) acceso al índice de segundo nivel en correspondencia con la tabla relativa a las segundas letras; búsqueda secuencial hasta encontrar la letra igual a la segunda letra de la denominación buscada; el puntero direcciona al subarchivo del archivo diccionario;



c) acceso al subarchivo del archivo diccionario y lectura secuencial de las denominaciones hasta encontrar la buscada.

Organización con acceso directo

También en este caso el archivo se contempla como una estructura secuencial y se organiza para contener los records en una posición preestablecida. La posición física de los records en el interior del archivo se determina aplicando un algoritmo de cálculo (fórmula de randomización) a la clave del record que hay que buscar o memorizar.

El archivo ha de poder alojar todos los posibles records cuya dirección se derive de tal cálculo y, por lo tanto, ha de ser lo suficientemente gran-

de como para poder memorizar más de los que efectivamente lo constituyen.
El gran inconveniente de esta organización es el desperdicio de espacio en el disco.

Resumiendo:

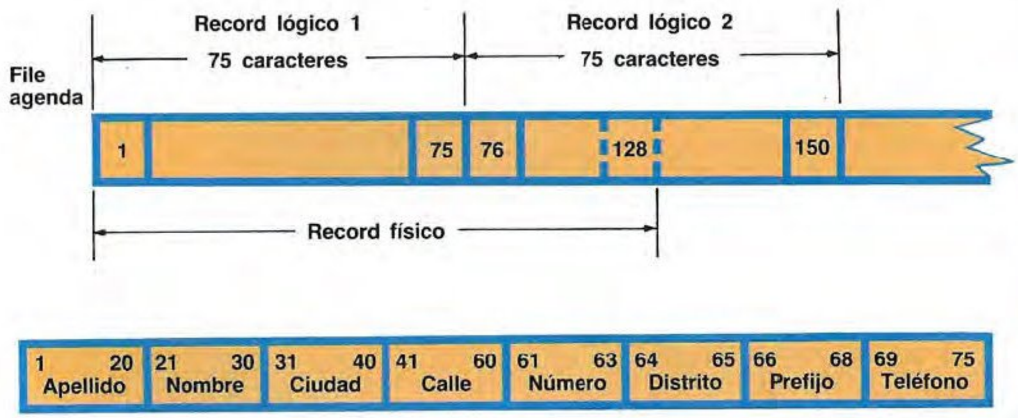
- **Los principales soportes magnéticos para la memorización de datos son: cintas, tambores, discos rígidos (fijos y móviles), diskettes (floppy).**
- **El conjunto de datos homogéneos relativos a un determinado tema se denomina archivo o file.**
- **En función de las operaciones de lectura y escritura, los files se dividen en records físicos, y en records lógicos a efectos de programación.**
- **Un record lógico puede ocupar varios records físicos o, viceversa, un record físico puede contener varios records lógicos, según la longitud de los campos.**
- **Los records lógicos se dividen en campos, cada uno de los cuales contiene un dato determinado.**

A título de ejemplo, apliquemos los conceptos expuestos a la estructuración de un archivo que contiene una agenda telefónica.
Los datos a memorizar son:

1 / Apellido	20 caracteres
2 / Nombre	10 caracteres
3 / Ciudad	10 caracteres
4 / Calle	20 caracteres
5 / Número	3 caracteres
6 / Distrito postal	2 caracteres
7 / Prefijo provincial	3 caracteres
8 / Número de teléfono	7 caracteres

Para cada dato se parte de una hipótesis de longitud en número de caracteres. El archivo contendrá sólo los datos mencionados, por lo que precisa un solo file. Archivos más complejos, con estructuras de datos diversificadas (es decir, de distintos tipos), se componen de varios files, cada uno de los cuales contiene uno de los tipos de datos. El record lógico está constituido por la suma de los caracteres ocupados por cada uno de los 8 datos (campos), y tiene, por tanto, una longitud de 75 caracteres. En el gráfico inferior se esquematiza la estructura propuesta, mostrándose un record físico, cuya longitud depende del tipo de máquina, ya que está determinada por el hardware del sistema. Hemos adoptado aquí una longitud de 128 caracteres, que es la más habitual en los microordenadores y ordenadores personales.
Como puede verse, un record físico contiene más de un record lógico; por lo tanto, si hay que

ESTRUCTURA DEL FILE AGENDA



Posición de los campos en el record lógico

SOLUCIONES DEL TEST 6

1 / c: la memoria de masa es de tipo permanente y puede conservar los datos por tiempo indefinido.
La memoria de la máquina se usa sólo para efectuar las elaboraciones de una parte de los datos.
Las salidas (datos elaborados) se transfieren nuevamente a las memorias de masa, ya sea para su conservación o para dejar libre la memoria del programa, que de esta forma se puede dedicar a la resolución de otro problema.
La técnica de «apoyar» los datos intermedios en un disco (el disco es la más rápida memoria masiva de acceso directo) se utiliza mucho en los ordenadores que trabajan a la vez para varios usuarios.
Si un usuario hace una pausa, sus datos se transfieren a un disco para dejar sitio a otra tarea.
El programa (o parte) interrumpido será reemprendido en cuanto disminuya el volumen de trabajo en la unidad central.
Todo este movimiento de datos y programas entre la memoria de la unidad central y la memoria de masa es gestionado automáticamente por el Sistema Operativo y permite optimizar el aprovechamiento de los recursos del sistema.

---

2 / En los formatos más pequeños, unos 250.000; en los mayores, 1.200.000. Las páginas de esta enciclopedia, cuando están escritas por completo, contienen dos columnas de 50 líneas cada una. En cada línea hay unos 45 caracteres, por lo que una página llena de texto contiene unos $2 \times 50 \times 45 = 4.500$ caracteres. El espacio ocupado por fotos y gráficos es aproximadamente un tercio del total, luego la obra contiene una media de 3.000 caracteres por página; un fascículo tiene 20 páginas, o sea $20 \times 3.000 = 60.000$ caracteres. En un diskette de 1.200.000 caracteres caben, por tanto, 20 fascículos, y toda la obra cabe en 5 diskettes de pocos gramos de peso, con un volumen equivalente al de unas pocas hojas.

---

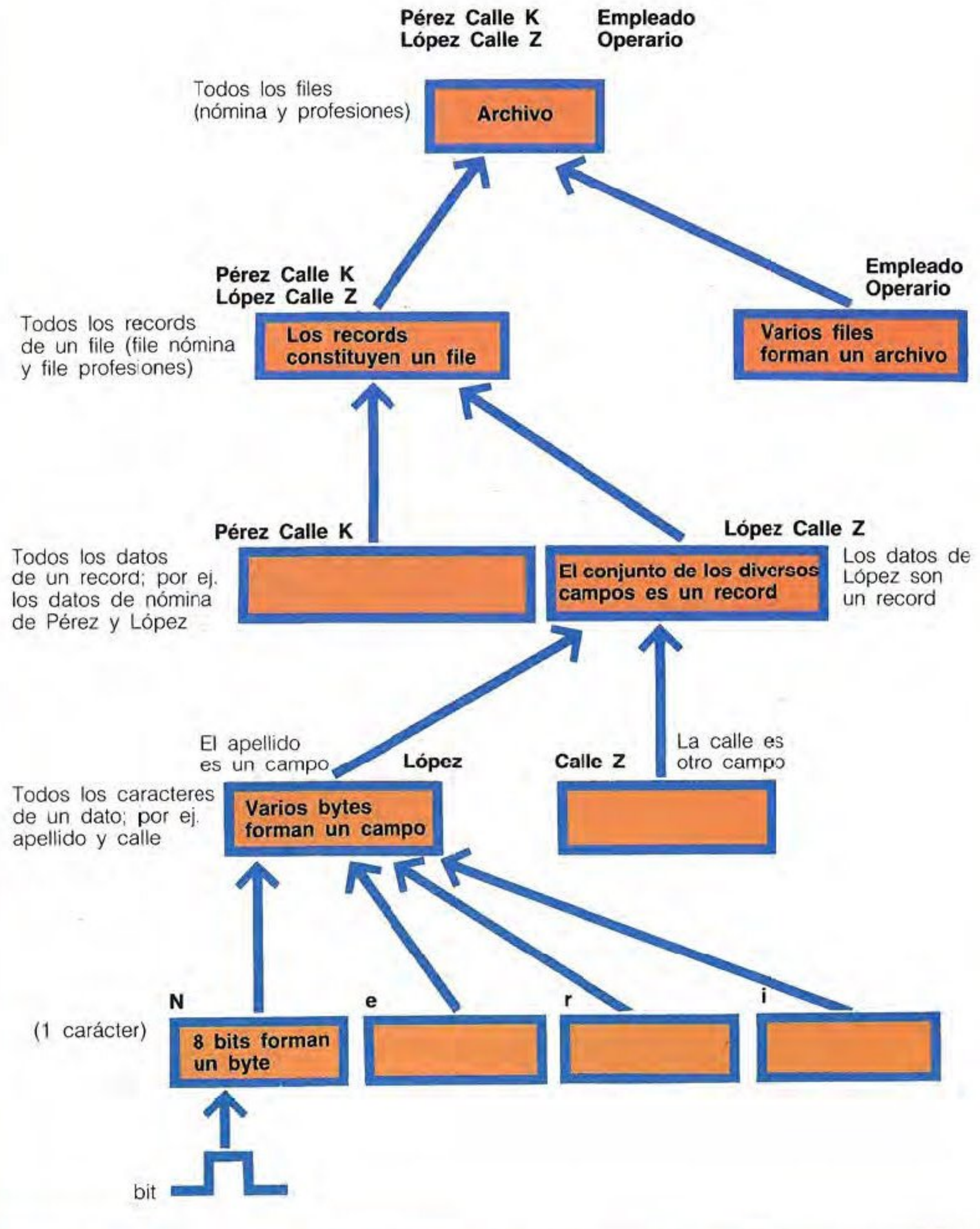
3 / a: los files son zonas de la memoria masiva que recogen bajo un mismo nombre (nombre del file) datos homogéneos.
b: los records pueden ser físicos o lógicos. Los records físicos constituyen la estructura lógica.
c: los campos son las zonas que contienen los datos en que están divididos lógicamente los records de los files.

---

4 / Se usan para hacer una copia de reserva de los archivos en disco. Debido a su índole secuencial, son dispositivos poco adecuados para contener archivos de uso constante. Por el contrario, los discos son de acceso directo y permiten altas velocidades de intercambio de datos.

5 / Bit, byte, campo, record, file, archivo. En el gráfico se esquematiza el orden jerárquico.

ORDEN JERARQUICO DE UN ARCHIVO



leer, por ejemplo, el record lógico 1, se transferirá a la memoria una parte del lógico 2; será tarea del Sistema Operativo (es decir, el conjunto de programas que se ocupan de las funciones básicas del sistema) separar luego los datos útiles de los no pedidos.

Como se ha dicho, el Sistema Operativo es un conjunto de programas básicos, que efectúa todas las funciones accesorias reales, como las de gobierno de las unidades de disco, de la pantalla, de la impresora y del teclado. La traducción de instrucciones de alto nivel (por ejemplo, en Basic) a microinstrucciones (lenguaje máquina) la realiza el Sistema Operativo.

En realidad, los mecanismos funcionales son más complejos y se comentarán en el capítulo dedicado a los Sistemas Operativos.

Organización de archivos en los microordenadores y ordenadores personales

En los grandes sistemas y en los miniordenadores más evolucionados, hay programas de gestión de archivos ya listos, que evitan al programador la tarea de entrar en el detalle de la búsqueda o la ordenación de los datos. Estos programas constituyen un «interface» entre el programa de aplicación, escrito por el programador, y la gestión de los archivos. Para realizar una función cualquiera, por ejemplo una búsqueda, basta con indicar, mediante órdenes, lo que se desea obtener; estos programas especiales se encargan de interpretar la instrucción, así como de activar el proceso de elaboración correspondiente.

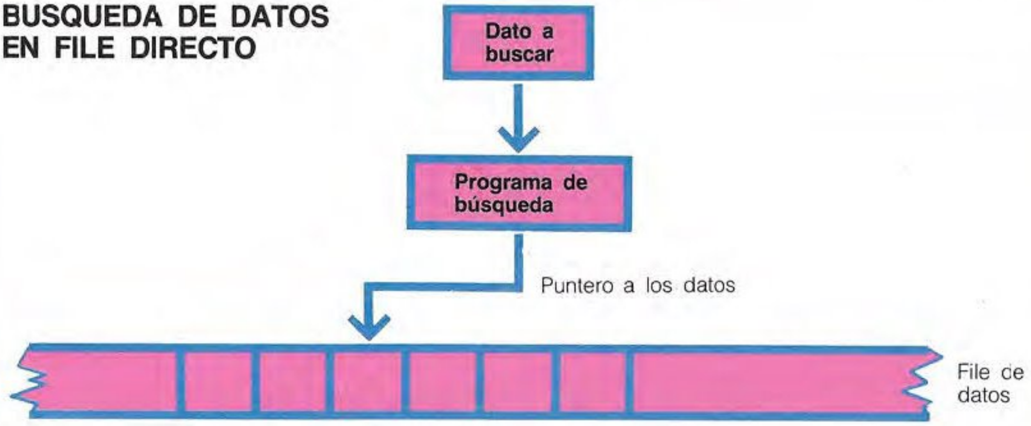
En los sistemas más pequeños (microordenadores y ordenadores personales) este tipo de software no existe, salvo en raras excepciones, y tiene, en cualquier caso, un costo elevado en relación al de la máquina; además, presenta algunos inconvenientes debidos a lo siguiente: los programas de gestión interna han de ser generalizados para que puedan cubrir la casi totalidad de aplicaciones; la forma generalizada lleva inevitablemente a ocupar mucho espacio tanto en diskettes como en la memoria, y en los sistemas más pequeños el espacio (es decir, las memorias) no es, desde luego, amplio.

Terminales en fase de prueba.



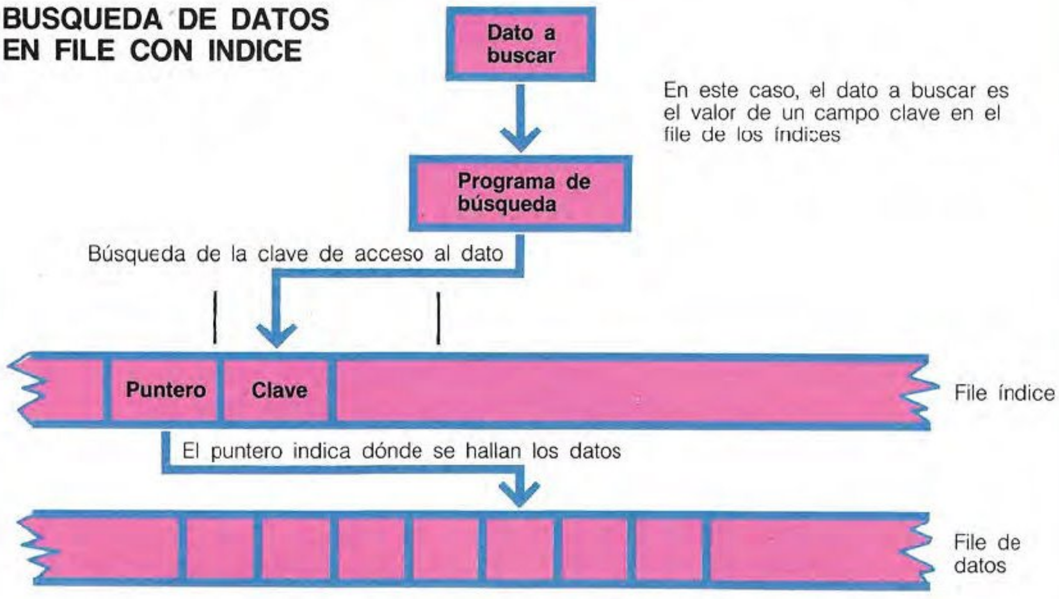
C. O'Rear/West Light-Grazia Neri

BUSQUEDA DE DATOS EN FILE DIRECTO



La búsqueda se efectúa directamente en el file de datos; la salida del programa de búsqueda indica («apunta») dónde se hallan los datos buscados

BUSQUEDA DE DATOS EN FILE CON INDICE



En el caso de estructura con índices se busca antes el valor de la clave en el file índice y luego, de ahí, se obtiene el puntero al dato, que reside en otro file

En algunas aplicaciones conviene escribir, además del programa de aplicación, también las partes necesarias para la búsqueda y ordenación de los datos, no generalizando, sino apuntando a la solución del problema específico.
De esta forma se puede conseguir un notable ahorro en la ocupación tanto de los diskettes como de la memoria.
Los dos principales métodos de búsqueda son los denominados **búsqueda en serie** y **búsqueda binaria** (o **dicotómica**). Pueden aplicarse directamente al file que contiene los datos, o bien al file de los índices. En este último caso, tras haber hallado el valor de la clave en el file de los índices, hay que utilizar el puntero contenido en él para extraer los datos del file correspondiente.
Los dos procedimientos de búsqueda en file directo o en file con índice se esquematizan en el gráfico superior.
Antes de describir los métodos de búsqueda, ilustremos el uso de las principales clases de

files utilizados en los microordenadores y ordenadores personales:

- **Files directos (random)**
- **Files secuenciales**
- **Files con índice**

Files directos (random)

En esta clase de file, como hemos dicho, la dirección del dato es el número del record que lo contiene: los datos del «grupo 1» se hallan en el record 1, los del «grupo 2» en el record 2, etc.
El método de acceso directo (DAM) se denomina también método random, y permite la escritura y/o lectura de un record especificando su posición física en disco o su posición absoluta o relativa en el file.
Un record puede leerse, pues, directamente conociendo el número del record mismo (posición física en el disco) o la posición relativa respecto a algunos datos del record.
En el direccionamiento relativo, el número del record puede usarse como clave del record mismo. La posición del archivo en la que buscar el dato (número del record) ha de ser establecida por el programa de aplicación; será, pues, tarea del programador alojar un determinado dato en un determinado record, y el programa de aplicación sólo podrá encontrar el dato direccionando el record correspondiente.
En la gestión de una agenda telefónica, la asociación entre el dato y el número del record puede ser el orden en que se introducen las señas: las primeras señas introducidas se alojan en el record 1, y así sucesivamente. Para encontrar los datos en el archivo habrá que remitirse al número de record correspondiente. Puesto que los datos se memorizan por orden cronológico de introducción, sin ninguna forma de ordenación, cualquier búsqueda implica la lectura de todo el file. Por ejemplo, si buscáramos todas las señas que tienen ciudad = MADRID tendríamos que leer todo el file, puesto que no sabemos dónde comienzan y dónde terminan las señas de Madrid.
Para eliminar este inconveniente, que implica notables aumentos en los tiempos de elaboración, antes de proceder a la selección de la ciudad Madrid conviene ordenar el file de datos con las ciudades por orden alfabético. De esta forma, en cuanto el dato adquiere un valor «mayor» que el valor MADRID (por ejemplo, MÁLAGA) se puede interrumpir la búsqueda, puesto que sabemos ya que entre los records siguientes no existe finalmente ninguno con ciudad = MADRID.
El proceso, desde el punto de vista lógico, está esquematizado en el gráfico de la página contigua. El diagrama de flujo es meramente indicativo; los métodos de búsqueda se expondrán detalladamente más adelante.
La mayor ventaja de esta clase de file está en el acceso directo a los records. La lectura o escritura de un record cualquiera es libre; así, se puede acceder al record 100 antes que al 50. Por esta característica, los files directos se denominan **files de acceso random** (al azar).

Files secuenciales

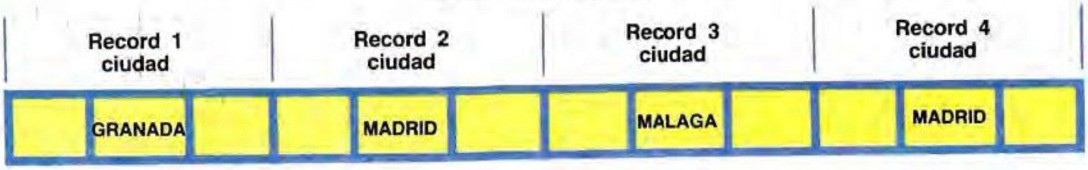
Estos files están ordenados secuencialmente por número de record, y no se puede acceder, en lectura o escritura, a un record cualquiera sin leer o escribir antes todos los records anteriores. Se utilizan para memorizar datos en cantidad fija y que hayan de ser leídos o escritos siempre todos. Un ejemplo típico son los files que contienen tablas.
Los records son leídos y/o escritos uno tras otro en secuencia física. Siempre que se crea un nuevo record, se graba al final del file. Para insertar un record en medio del file hay que reescribirlo entero, colocando el nuevo record en la posición deseada. La lectura de un record puede efectuarse sólo recorriendo secuencialmente el file hasta encontrar el record buscado. Los records se pueden actualizar o no, según los recursos del método de acceso.

Files con índice

Son files random provistos de índice, que puede residir en otro file o formar parte del file de datos mismo. Dada su importancia en las aplicaciones, conviene examinarlo con más detalle.
Como hemos visto, hay dos clases principales de estructuras de datos asociadas a un file con índice: el file índice y el file de datos (ver gráfico de pág. 250). El file índice es una secuencia de records que contiene la clave del record de datos y un puntero al mismo. La clave identifica unívocamente el record de datos. Hay también índices secundarios o alternativos que suministran vías de acceso variadas al file de datos.
También es posible tener la misma clave repetida varias veces en un file índice. Por ejemplo, si el nombre de una persona se usa como clave, puede haber varias personas con el mismo nombre. Con objeto de conservar el acceso uní-

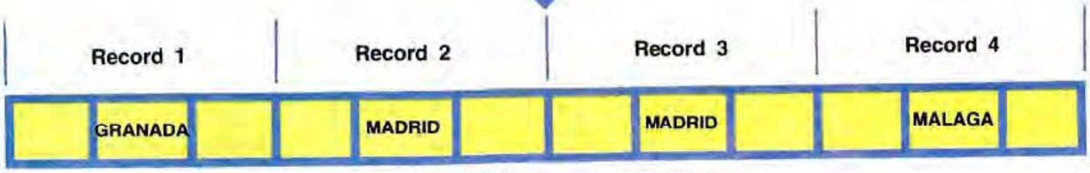
DIAGRAMA DE FLUJO QUE MUESTRA LA INTERRUPCION DE LA BUSQUEDA EN UN FILE ORDENADO

File de datos originario



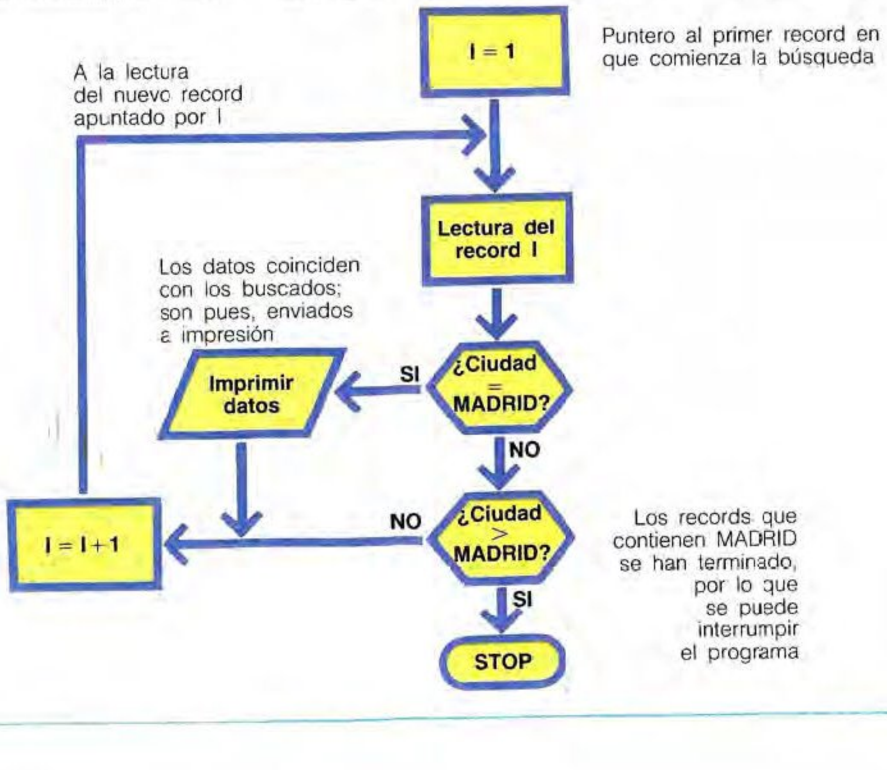
En el file originario, los datos están colocados por orden de introducción; en la fase de búsqueda, por ejemplo por ciudad, hay que leer todo el file

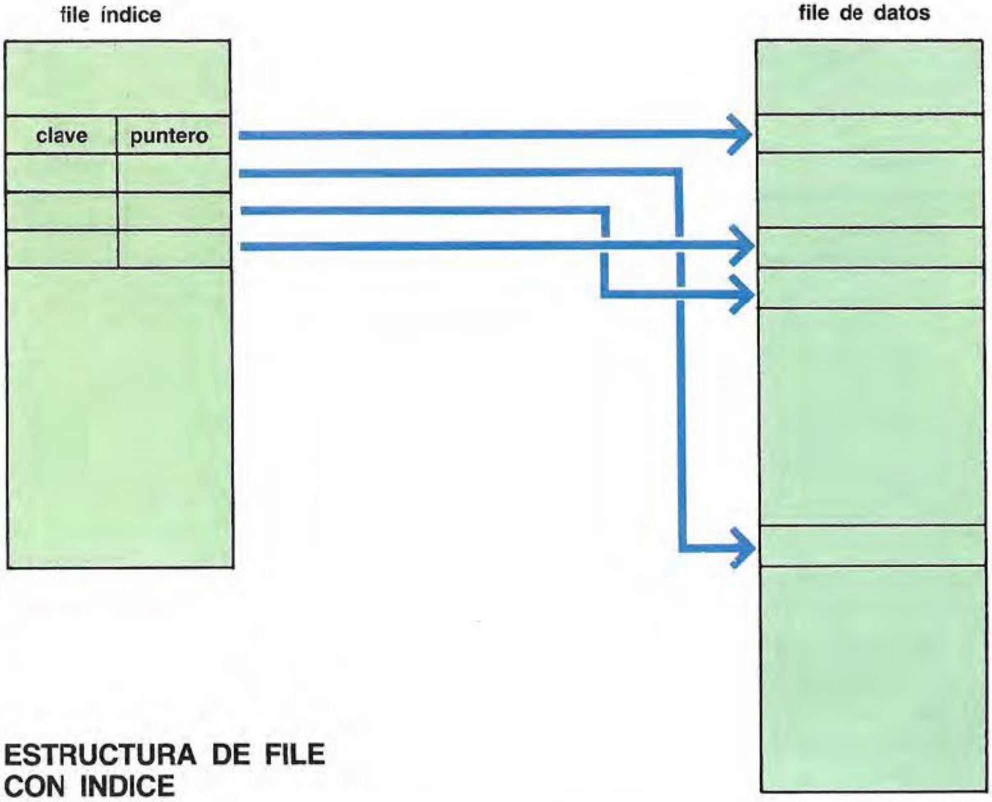




Nuevo file con los campos ordenados

En este caso, la búsqueda puede terminar en el record 4, puesto que en él hay un dato (ciudad = MALAGA) cuyo valor es mayor del que estamos seleccionando (MADRID); estando el file ordenado con los nombres de las ciudades por orden alfabético, no se pueden encontrar otros records MADRID después del 4





ESTRUCTURA DE FILE CON INDICE

voco a todos los records del file, es necesario, en casos como este, especificar el número correspondiente del record en el conjunto de records con la misma clave.

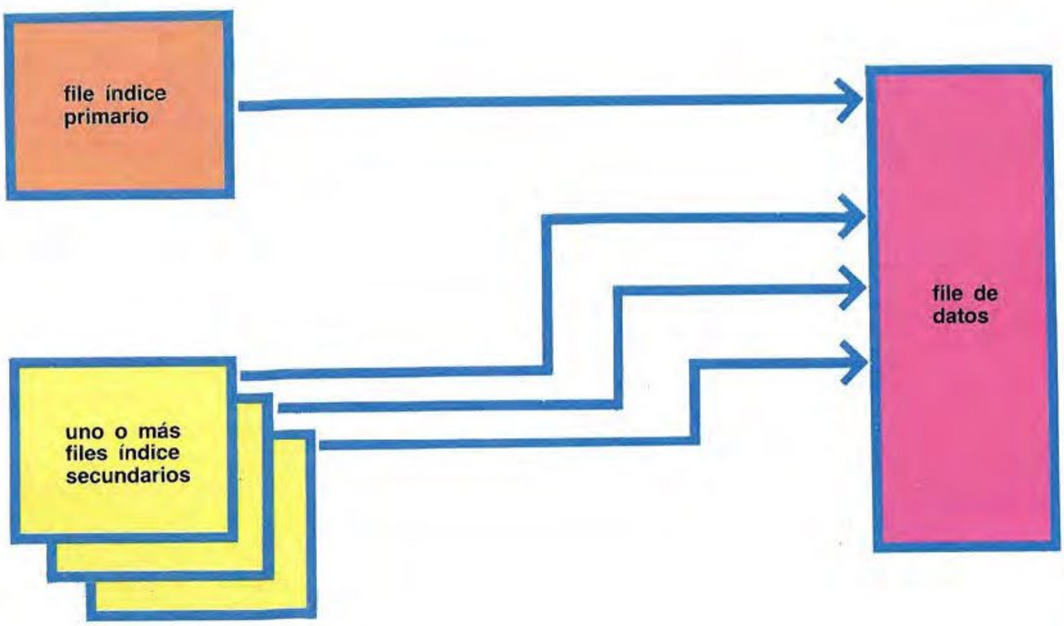
El file de datos contiene los records de datos. Puede estar organizado en base al valor de la clave, o bien de forma distinta, según la técnica de implementación. Los índices secundarios permiten el acceso a los records mediante claves distintas de la principal (primaria). Por ejemplo, los records pertenecientes a un file de clientes pueden ser leídos especificando el número de código del cliente o bien su nombre.

El índice primario se formará, pues, usando el número de código como clave; el índice secundario utilizará el nombre como clave secundaria. De esta forma tenemos dos accesos al file distintos e independientes (ver gráfico de pág. 151). A veces puede ser necesario acceder a los records usando como clave la combinación de varios valores. Con tal objeto, dos o más datos se combinan (o concatenan) para formar el valor de la clave.

Por ejemplo, una empresa con varias sucursa-

les podría desear mantener lógicamente separados los clientes de las distintas sucursales con objeto de hacer fácilmente las cuentas para cada una de ellas.
Si la clave del record está formada por el código de la sucursal y el número del cliente, todos los clientes de una misma sucursal quedan agrupados y pueden ser tratados de forma independiente con respecto a otras sucursales. Ello no excluye, sin embargo, la posibilidad de una gestión global de todo el file de clientes.
Los files con índice son los de mayor versatilidad de gestión, puesto que permiten efectuar operaciones de búsqueda optimizadas con respecto al dato que se desea extraer. Son, sin embargo, las estructuras de archivo más difíciles de gestionar. Los programas de gestión de los files con índice son bastante complejos, y requieren experiencia por parte del programador.
Dada la importancia práctica de estas estructuras, en el próximo apartado expondremos el sistema de gestión de files con índice ISAM, puesto a punto por Olivetti e implementable en microordenadores y ordenadores personales.

ESTRUCTURA DE FILE CON INDICES SECUNDARIOS



La oficina computerizada

La oficina es, todavía hoy, el corazón de toda empresa industrial o comercial; lo demuestra el hecho de que, a partir de los años cincuenta, el porcentaje de la fuerza de trabajo ocupada en las industrias manufactureras ha disminuido drásticamente, mientras que ha aumentado enormemente el número de personas que trabaja en oficinas. En Estados Unidos, el número de personas empleadas en el sector de la elaboración de la información es hoy superior al de cualquier otro tipo de ocupación. En las oficinas se manejan tres tipos fundamentales de datos: números, diagramas y palabras. La primera fase de la revolución que ha llevado a la posibilidad de eliminar el documento sobre soporte de papel y que, en principio, podemos considerar hoy completada, introdujo la elaboración electrónica de los datos, a la que siguieron innovaciones radicales en el sector del grafismo y el denominado proceso de palabras, efectuado también por ordenador. Los modernos ordenadores han suprimido gran parte del tedioso trabajo de manipulación de cifras, permitiendo a los contables y directores de ventas aplicar principios de proceso de datos y de análisis estadísticos inconcebibles antes de ahora.

La enorme difusión de calculadoras y microprocesadores de costo moderado ha determinado claramente las líneas evolutivas en el campo del cálculo automático. Los ordenadores personales tienen un costo que es una modesta fracción del de los grandes ordenadores, y aunque no pueden compararse a sus hermanos mayores en cuanto a velocidad, capacidad de memoria y amplitud de análisis, son perfectamente adecuados para las funciones típicas del trabajo de oficina como, por ejemplo, la compilación de los libros principales de contabilidad.

La creación de procesadores de palabras ha permitido extender las posibilidades de la elaboración automática a los textos escritos. La técnica moderna puede desvincular las palabras de su tradicional soporte de papel para transferirlas al mundo de la microelectrónica. No pasará mucho tiempo antes de que "ejércitos" de esclavos de la máquina de escribir sean liberados de su fatigoso trabajo.

Gracias a los procesadores de palabras, un empleado puede transmitir un comunicado codificando su texto, asignándole los pertinentes niveles de prioridad y confidencialidad y marcando en un teclado la dirección electrónica del destinatario. Al llegar a destino, el mensaje es insertado entre los otros datos que llegan, para su introducción en la memoria del sistema, según su grado de prioridad. El destinatario puede seleccionar, mediante un teclado especial, y leer en una pantalla cualquiera el texto, descartando el material inútil y archivando los demás datos en la memoria del ordenador, junto con eventuales cambios y comentarios. Ningún papel pasa por la oficina, ningún papel es conservado o tirado y no hace falta ningún intermediario para el transporte físico del mensaje.

Recordando las siglas de referencia del documento, éste podrá ser llamado automáticamente a la memoria, y el procesador de palabras lo hará aparecer nuevamente en el monitor, permitiendo al operador introducir correcciones antes de dar a la máquina las instrucciones que harán producir un número cualquiera de copias perfectas. Todo esto sucede mucho más rápidamente de lo que podría hacer una mecanógrafa. Hoy las copias de un documento a enviar se imprimen en papel, pero el mismo mensaje puede ser enviado por la memoria del procesador de palabras directamente a la pantalla de un terminal conectado mediante línea telefónica.

El establecimiento de una red de telecomunicaciones dotada de posibilidades de memorización llevaría casi a la total eliminación del uso del papel en las oficinas. Las aplicaciones podrían diversificarse aún más gracias al empleo de las conexiones mediante fibras ópticas, que transmiten impulsos de láser codificados, o de los satélites de telecomunicaciones.

Los editores científicos especializados ya están creando centros de elaboración destinados a evitar las pérdidas de tiempo y los costos ligados a la compilación normal y la preparación de la información científica por escrito. En pocas palabras, el autor tiene la posibilidad de hacer un «manuscrito» en cinta magnética y enviarlo por vía telefónica a la editorial, donde el texto es introducido en un procesador de palabras y memorizado en un archivo. Posteriormente, cualquier persona interesada y habilitada para hacerlo puede seleccionar el texto y hacer que aparezca en la pantalla de su propio lugar de trabajo, sin que nunca haya sido impreso en papel. En numerosas bibliotecas universitarias el sistema ya está en uso para archivar en cinta extractos de revistas técnicas y científicas.

También las redacciones de los periódicos han

cambiado radicalmente. Provistos de monitores y «consolas» conectadas con el ordenador central, cronistas, editores y componedores pueden componer un periódico entero –textos, figuras y titulares– operando directamente a nivel de memoria del sistema de elaboración. De ser necesario, el proceso podría desarrollarse por completo sin utilizar ni una hoja de papel. La imagen electrónica del periódico terminado, realizada en el seno del ordenador, se utiliza luego para preparar la máquina de impresión normal. Falta sólo un pequeño paso para llegar al momento en que los datos se transmitirán directamente a pantallas especiales instaladas en los transportes públicos o a los aparatos de televisión de las casas. Los lectores podrán así recibir las últimas noticias durante la comida, seleccionando, eventualmente, algún artículo para conservar en cinta, y continuar la lectura de los diversos sucesos y los respectivos comentarios durante el viaje hacia el puesto de trabajo.

El concepto de «estación de trabajo» prevé la realización de pequeñas unidades para una o dos personas en el ámbito de una gran oficina sin paredes divisorias fijas. Cada unidad está formada por los materiales y los instrumentos necesarios para el desarrollo del trabajo, e incluye monitores, «consolas» e interfaces de acceso a la memoria del ordenador y al sistema de telecomunicaciones de la empresa. La zona reservada a cada estación de trabajo está delimitada por una división a la altura de un hombre, que garantiza cierta reserva y tranquilidad.

Estudios ergonómicos profundos, realizados también gracias a los ordenadores, han permitido determinar los valores más favorables de varios factores ambientales que afectan al trabajo en las oficinas, entre ellos el nivel de iluminación, las distancias entre los escritorios y el calentamiento producido por los aparatos. Por ejemplo, la mayor parte de los ordenadores funciona sin hacer ruido; sin embargo, algunos elementos emiten sonidos de alta frecuencia, inaudibles para el oído humano, pero que pueden producir molestias. También estos aspectos han de ser evaluados cuidadosamente.

Es evidente que en una oficina sin papeles, la autoridad de la firma humana ha de ser sustituida por documentos legales equivalentes. Se ha de cuidar, además, la confianza del público en la infalibilidad de la memoria electrónica, tomando medidas preventivas para evitar que en los ordenadores se introduzcan errores humanos



Brecht Einzig Ltd.



Lyndsay Westhead Architects/Environment

La utilización de ordenadores ha replanteado la solución de la oficina «abierta».

que pudieran acumularse con consecuencias desastrosas. Recientemente, el error de un empleado le causó a una fábrica inglesa una pérdida de medio millón de dólares. Los problemas de este tipo serán resueltos en breve.

Los problemas planteados por la envergadura de las telecomunicaciones públicas e internacionales, por los archivos de datos y las tareas en distintos idiomas se resolverán mucho antes que la reestructuración de los modelos de trabajo: archiveros, mecanógrafos y recaderos tendrán que ser desplazados hacia otras tareas, y su número disminuirá inevitablemente.

Los directivos tendrán que aprender nuevas nociones y adaptarse al nuevo ambiente psicológico, que ya ofrece algunas sorpresas. El personal que trabaja en las oficinas avanzadas descubre una cierta difuminación de las diferencias entre las jerarquías habituales: las comunicaciones basadas en la electrónica tienden a reducir la autoridad de los superiores y a sustituirla por una especie de «respeto» hacia la autoridad de la pantalla visualizadora.

Gestión de un file con índice (sistema ISAM)

Un sistema de gestión de files con índice permite el acceso a los records de los files tanto en forma secuencial como directa (random), en función de una clave (o dato identificador) de cada record de datos.

Ambos métodos ofrecen ventajas para tipos de aplicación concretos. La estructura de file con clave se muestra especialmente útil donde es necesario acceder a los records tanto en forma secuencial como al azar.

Esta estructura de file con clave es mucho más flexible que otros tipos de organización, y resulta además especialmente adecuada para aplicaciones sofisticadas, tanto interactivas como en tiempo real.

Un campo muy adecuado para la utilización de files con clave es la consulta de archivos, es decir, la posibilidad de leer la información contenida en un record utilizando sólo la clave.

Por ejemplo, en un sistema de gestión de almacén hay que conocer las existencias de un determinado artículo antes de efectuar eventuales pedidos. Mediante la utilización de un file con clave, definiendo como clave el código del artículo, un programa puede solicitar del operador el número de código, leer el record correspondiente y visualizar las existencias del artículo seleccionado.

Aprovechando los índices secundarios, se puede asignar una segunda clave al artículo (por ejemplo, su descripción); de esta forma, si no se conoce el número de código, se puede suministrar la descripción del artículo para obtener el record deseado, o varios records con la misma descripción. Un segundo ejemplo de utilización eficaz de los files con clave es el control de datos: en una estructura de file con clave, el valor de un dato puede ser usado como clave.

En este caso, el valor suministrado por el operador puede ser comprobado mediante una lectura random con clave en la estructura del file en el que el dato se utiliza como clave. Si la clave no existe, el operador es informado inmediatamente del error, y puede suministrar otro valor. Por ejemplo, en un sistema de gestión de sueldos, el record de las horas de trabajo semanales contiene también la matrícula del empleado.

La introducción de ordenadores, además de haber revolucionado el concepto tradicional de archivo, le ha aportado mayor orden y rapidez de búsqueda.



C. O'Rear/West Light-Grazia Neri

El número de matrícula es introducido por el operador, usándolo como clave para acceder al record del empleado. Si la clave no es encontrada, el operador es informado y puede introducir el valor correcto.

Operaciones de acceso a los datos

Para leer un record en un file con índice (a partir de ahora lo llamaremos también file con clave) se pueden utilizar cinco métodos:

- **Lectura random por clave.** Con este método hay que suministrar un valor de clave y se obtiene un puntero al record asociado. Una vez especificada la clave, el método de acceso efectúa una búsqueda en el file índice para encontrar el correspondiente puntero al record. Si la búsqueda tiene éxito se obtiene el puntero; en caso contrario, se emite un código de error. La lectura random por clave es útil para acceder a y actualizar un file cuando se desea un record muy concreto y se conoce su clave.

- **Lectura genérica por clave.** Este método es similar al anterior, con la diferencia de que, si el record especificado no existe, se obtiene el puntero con el valor de clave que sigue inmediatamente.
  El empleo de este método con claves parciales ofrece la posibilidad de agrupar records de datos en secciones lógicas, y constituye un eficaz sistema para seleccionar y organizar records de datos.

- **Lectura secuencial por clave.** Este método ofrece la posibilidad de leer records en secuencia lógica por valores de clave, independientemente de la secuencia física de los records en el file.
  La lectura puede comenzar por el primer record lógico del file o desde cualquier otra posición.
  Este método puede usarse para acceder al record siguiendo una determinada secuencia, por ejemplo, por valor de clave, o bien junto con la lectura genérica, para leer grupos lógicos de records.

- **Lectura secuencial en orden físico.** Permite leer en el file de datos los records en orden físico, sin referencias al file índice.
  Este método es útil para aplicarlo a elaboraciones de datos en las que la secuencia de lectura de los records no es tan importante como el análisis de los datos, y reduce el tiempo de acceso a los datos mismos, eliminando la búsqueda en el file índice.

- **Lectura random por número de record.** Permite acceder a los datos mediante su número de record.
  Ello da acceso a un file de datos tanto de la forma DAM como directamente por clave.

También las operaciones de inserción y cancelación de datos pueden realizarse de distintas maneras.
Un sistema de gestión de file con clave ofrece la posibilidad de sacar datos y/o su clave de la estructura básica. Esto reduce el trabajo de programación requerido para la gestión de los records borrados, ahorra espacio en el disco y disminuye la necesidad de reorganización de los files.
Hay una adecuada función de cancelación que permite eliminar records de un determinado file. Esta función quita del file índice la clave asociada al record y deja el record mismo disponible para volver a utilizarlo cuando se desee añadir otros. En el caso de que se desee eliminar la clave asociada a un record conservando el record mismo, hay que usar la función de cancelación de clave. Esta función elimina la clave del file índice, pero no altera el record de datos.
Dicho record puede ser leído mediante acceso secuencial. La responsabilidad de eliminación del record de datos queda, pues, delegada en el programador.
La cancelación de clave es muy útil en aplicaciones donde se prevé acceder al record en tiempos sucesivos, como, por ejemplo, en la generación de archivos; también es útil para eliminar de los files índice secundarios las claves de los records borrados.
El ejemplo más significativo y útil de utilización de files con clave es la actualización de datos en tiempo real.
Un dato contenido en un record puede mantenerse al día respecto a la evolución del programa aplicativo; se convierte, así, en una información significativa en tiempo real.
Normalmente, esto sucede conjuntamente a las fases de lectura de record.
Por ejemplo, en los medios bancarios, las cantidades que entran y salen de una cuenta co-

rriente se pueden registrar directamente mediante una lectura con clave (número de cuenta), añadiendo o restando la suma correspondiente y reescribiendo el record en el file de las cuentas corrientes. Cada lectura o transacción sucesiva reflejará el nuevo estado de cuenta independientemente del tiempo transcurrido.

Estructura general del file ISAM

El ISAM utiliza dos estructuras de datos: files índice y files de datos. Pueden coexistir varios files índice si se utiliza una estructura de claves secundarias, es decir, con posibilidad de definir varias claves secundarias, o sea, con posibilidad de definir varias claves de acceso al record de datos.

Los files de datos ISAM se memorizan según el formato Basic y son compatibles con todas las funciones Basic.

Los programas de utilidad como «Sort» y «Merge» pueden, pues, utilizarse sin ninguna modificación.

El «Sort» es un programa para la ordenación de datos; el «Merge» es la operación con la que se «unen» varios files de datos para formar un único archivo residente en un solo file.

Hay que señalar, sin embargo, que dichos programas provocan la destrucción de los lazos lógicos entre file de datos y file índice.

El file de datos es un file Basic de acceso directo (random) y puede, por tanto, gestionarse mediante las instrucciones Basic relativas a dicha clase de file. El file índice está formado por records que contienen cada uno un determinado número de claves; a cada clave le corresponde un número de record.

Las claves duplicadas requieren un tratamiento especial, puesto que el mecanismo de búsqueda es ligeramente distinto. Esto significa que hay dos clases de files índice: los que admiten claves duplicadas y los que sólo admiten claves únicas. En cualquier caso, las claves no se memorizan en secuencia física, sino que están conectadas mediante punteros que las mantienen en orden alfanumérico.

Esta secuencia se gestiona mediante una estructura «en árbol».

La estructura en árbol utilizada por el ISAM se denomina **B-tree**.

La estructura B-tree

Antes de ver con detalle la estructura B-tree utilizada por el ISAM, conviene examinar una forma más simple con objeto de comprender mejor su funcionamiento.

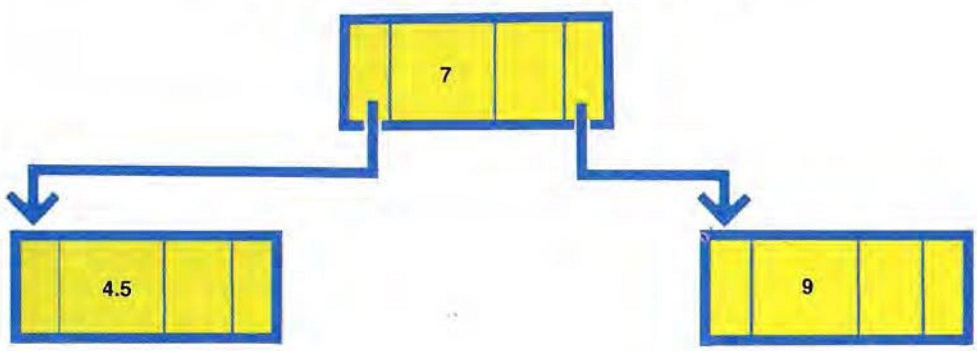
En una de las formas del B-tree, cada clave ocupa un nudo del árbol y tiene asociados dos punteros: uno a la clave inmediatamente anterior y otro a la siguiente; estas dos claves están en el nivel inmediatamente inferior.

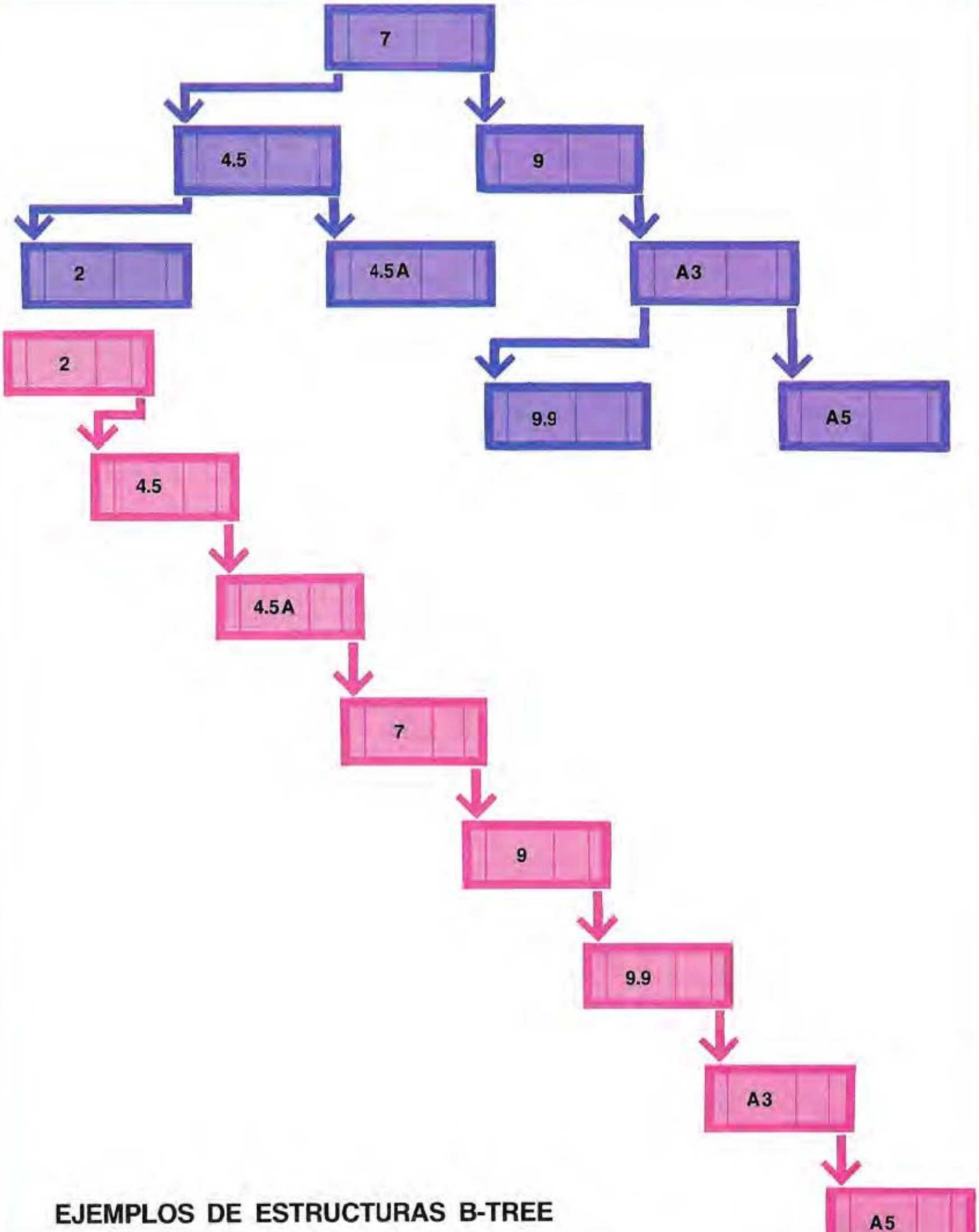
Esta estructura se conoce también con la denominación de «árbol binario».

El gráfico inferior ilustra la estructura B-tree tras añadir las claves «4.5» y «9» a la preexistente clave «7».

Cada vez que se añade un record al file, su clave se compara con el valor de la clave superior del árbol; este valor se denomina «raíz» del árbol. Si la clave del record a insertar es mayor que la de la raíz, aquélla se compara con la asociada al puntero de la derecha de la raíz; análogamente, si la clave del record a insertar es me-

EJEMPLO DE ESTRUCTURA B-TREE





EJEMPLOS DE ESTRUCTURAS B-TREE

nor que la de la raíz, se compara con el nudo «apuntado» a la izquierda. Se efectúan sucesivas comparaciones hasta alcanzar la posición correcta de la clave en el nivel más bajo del árbol. Por ejemplo, añadiendo las claves 2, A3, A5, 4.5A y 9.9 a la estructura de la pág. 256, obtendremos la estructura del gráfico superior de esta página. Mientras el árbol permanezca equilibrado (el número de claves a la izquierda de un nudo es comparable al de la derecha), el tiempo de búsqueda de una clave resulta entonces optimizado.

Por ejemplo, la clave A5 del gráfico superior de la pág. 257 puede alcanzarse tras haber examinado los nudos 7, 9 y A3, o sea después de tres comprobaciones. En el caso de que las claves se inserten en secuencia alfanumérica tendremos la estructura del gráfico inferior de la pág. 257, en la que se necesitan siete comprobaciones para alcanzar la clave en cuestión. Al objeto de evitar desequilibrios de este tipo, hace falta un algoritmo que, mediante ajustes (splitting) del árbol, lo mantenga equilibrado.

La estructura ISAM de árbol

El B-tree utilizado por el ISAM no es tan simple como el que acabamos de ver. El ISAM utiliza dos clases de records: los nudos índice y los nudos terminales.

Los nudos terminales están en el nivel más bajo del árbol y contienen todas las claves con los correspondientes números del record de datos. Los nudos índice ocupan todos los niveles del árbol, excepto el inferior, y ofrecen un camino para el acceso a los nudos terminales (ver gráfico inferior). Además, cada nudo terminal posee dos punteros, uno en el nudo terminal anterior y otro en el siguiente. Esto permite el acceso a todas las claves respetando la secuencia ASCII.

Puesto que la longitud del record en el file índice es fija, mientras que la de las claves es variable, también el número de claves por record resulta variable.

Número de files abiertos y files buffering

El usuario ha de definir el número máximo de files índice que pueden abrirse simultáneamente; hay una primitiva* en el ISAM que permite especificar este número, que ha de estar comprendido entre 1 y 15.

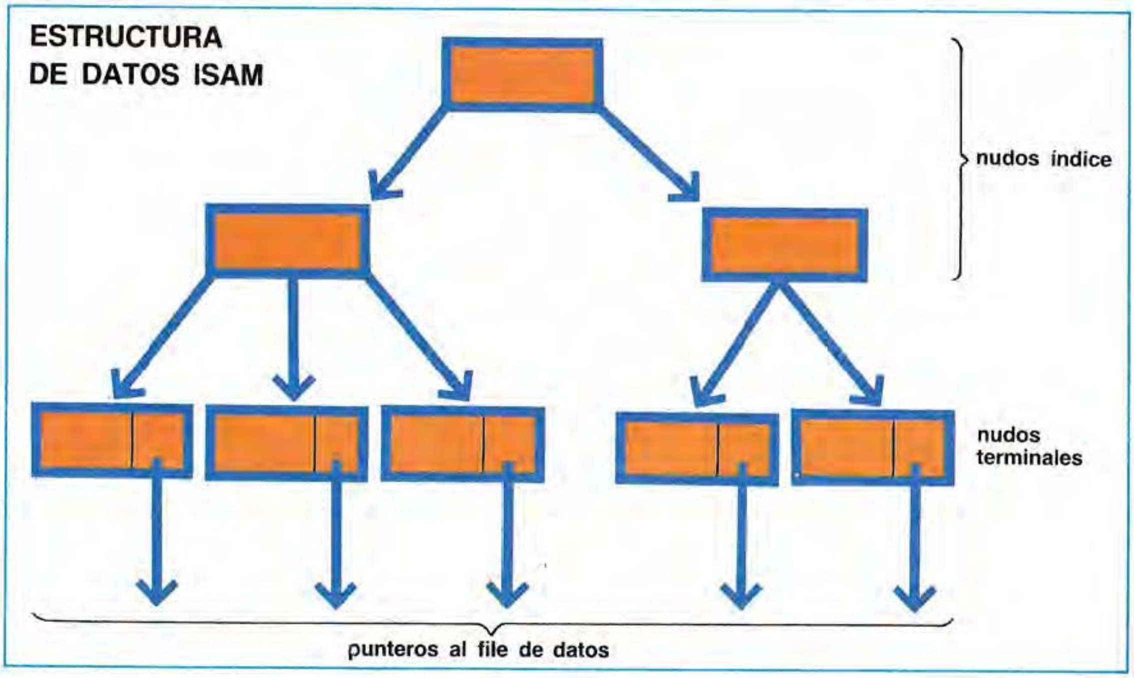
Los números asociados a los files índice utilizados en el ISAM han de ser distintos de los asignados a los files de datos; por ejemplo, si el ISAM ha sido preparado para un máximo de cuatro files (sean 1, 2, 3 y 4), los números disponibles para files distintos de los files índice son del 5 al 15.

Si un programa Basic utiliza los números 1, 2, 3 o 4, pueden producirse errores imprevisibles. Un método para eliminar problemas de este tipo consiste en numerar los files de datos en orden decreciente, empezando por el número más alto de files disponibles.

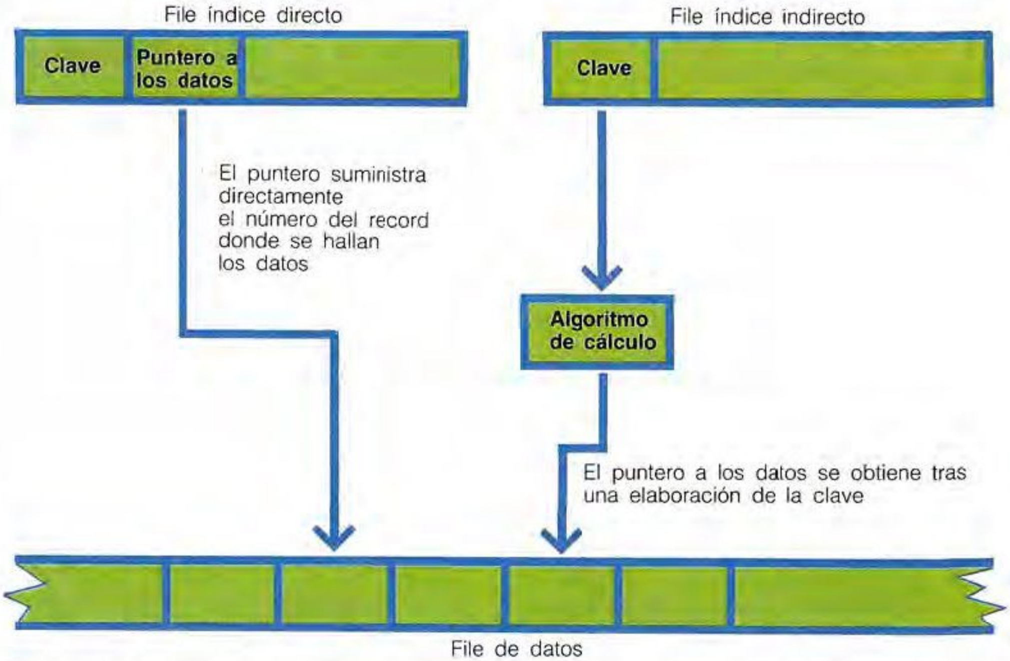
Los records del file índice son leídos mediante «buffers» de file (áreas temporales) alojados en la memoria del sistema. Este reserva automáticamente un mínimo de dos buffers para cada file, permitiendo el funcionamiento del algoritmo de compensación del B-tree en el file índice.

Sin embargo, el usuario tiene la posibilidad de aumentar el número de buffers, limitado sólo por

* Con el término «primitiva» se designa un programa básico (parte del Sistema Operativo) que realiza funciones de interface entre el usuario y la máquina.



COMPARACION ENTRE LA GESTION DE DATOS CON FILE INDICE DIRECTO E INDIRECTO



En el segundo caso, el índice ocupa menos espacio, puesto que no contiene el puntero a los datos; el valor del puntero se obtiene mediante un algoritmo de cálculo aplicado al valor mismo de la clave

la disponibilidad de memoria: a medida que aumenta el número de files, aumenta también la cantidad de memoria necesaria para su gestión (DBC, etc.) y decrece, pues, la cantidad de memoria disponible para los buffers.
Es aconsejable, por tanto, minimizar el número de files abiertos, dejando disponible la mayor cantidad de memoria posible para los buffers.
Aunque se recomiendan dos buffers por file índice, el ISAM funciona normalmente también con uno solo; en este caso, sin embargo, su eficiencia disminuye.
El ISAM está en condiciones de hacerlo porque contiene algoritmos que eligen los buffers disponibles para los files que han sido usados más recientemente, haciendo así el mejor uso posible del espacio del buffer. Es decir, cuando un file índice requiere un buffer adicional y todos los buffers son utilizados simultáneamente, se elige el último buffer que ha sido usado.

Direccionamiento indirecto

En los files con clave, la estructura más simple consiste en memorizar para cada clave el número del record (o de los records) donde se halla los datos correspondientes. Un método alternativo consiste en obtener la posición del record del valor mismo de la clave. El método es más complejo que el directo, pero permite ahorrar espacio, puesto que no es necesario memorizar, además de la clave, los números de los records que contienen los datos.
En el gráfico superior se esquematizan los dos sistemas: el primero, ya conocido, tiene un file índice que hace de puntero para obtener los datos; en el segundo (direccionamiento indirecto) el puntero a los datos se obtiene elaborando adecuadamente el valor de la clave.
Los métodos de cálculo de la dirección a partir de la clave pueden ser bastante complejos; a continuación se mencionan sólo los principales, aplicables a claves numéricas.

Subdivisión. La clave (numérica) se subdivide en un determinado número de grupos, los números obtenidos se suman entre sí y el resultado se toma como puntero a los datos. Por ejemplo, sea 537146 el valor de la clave; se divide en

tres grupos de dos cifras:

53, 71, 46

que sumados dan: 53 + 71 + 46 = 170; este es el valor del puntero a los datos. En la fase de introducción, los datos que tienen la clave 537146 se escribirán en el record 170. Viceversa, en fase de búsqueda, dada la clave, bastará con sumar los números de dos cifras que contiene para obtener la posición (record 170) de los datos. El número de grupos (y las cifras de cada grupo) en que se divide la clave antes de la suma, depende de cada aplicación.

Multiplicación. El método es análogo al anterior, con la única diferencia de que la operación final es el producto.

En el ejemplo anterior, la clave 537146 puede dividirse en dos grupos: 537 y 146. Su producto da 78402, que puede usarse como puntero a los datos, ya sea completamente (los datos se situarán en el record 78402), ya parcialmente, tomando, por ejemplo, sólo las tres cifras centrales: 840.

Multiplicación por 11. La clave se multiplica por 11 y el resultado se toma como puntero (también en este caso cabe utilizar sólo las tres cifras centrales).

Cuadratura. El puntero es una parte del cuadrado de la clave. Por ejemplo, con la clave 271 tenemos: $271^2 = 73441$, y tomando las tres cifras centrales, el puntero es 344. Las principales dificultades al utilizar estos métodos son la existencia de sinónimos y la distribución de los datos.

Cualquiera que sea el método adoptado (subdivisión, multiplicación, etc.), puede suceder que dos o más claves distintas den el mismo puntero. Por ejemplo, las claves 7421 y 5243, con el método de la subdivisión dan:

7421 = 74 + 21 = 95

5243 = 52 + 43 = 95

Tendrían, pues, el mismo puntero a los datos (record 95). El problema se resuelve reservando una zona del índice a los sinónimos, es decir, a las claves de distinto valor que suministran (debido al algoritmo de cálculo) punteros iguales. Naturalmente, los sinónimos han de tratarse de forma distinta que las demás claves. El segundo problema surge cuando el algoritmo de cálculo suministra valores demasiado concentrados en

La gestión de bancos de datos requiere imponentes memorias de masa.



Marka

una zona; como resultado, se tiene el file de datos lleno en una zona y vacío en otra. La solución consiste entonces en cambiar el algoritmo.

Ejemplo de aplicación: gestión del balance familiar

Para ilustrar las técnicas de creación y gestión de files vistas hasta ahora, desarrollemos un programa para la creación y gestión de un archivo con el balance familiar. El primer paso (análisis) consiste en determinar cuáles serán las salidas que necesitaremos y cuáles han de ser las entradas para obtener dichas salidas.

El plan de cuentas

Tratándose de un balance, la primera distinción se hará con respecto al tipo de «movimiento», es decir, distinguiendo entre ingresos y gastos; luego habrá que saber el tipo de ingreso (sueldo, renta, etc.) y el tipo de gasto (alquiler, luz, alimentación, ropa, etc.). La voz que especifica el tipo de movimiento se llama «causal»; la tabla que enumera las posibles voces de la causal es el «Plan de Cuentas». Resumiendo, tenemos las siguientes posibilidades:

Tipo de cuenta:
1 / Gastos
2 / Ingresos

Conceptos cuenta de gastos (10)	**Conceptos cuenta de ingresos (20)**
11 / Alquiler	**21** / Sueldo principal
12 / Luz y agua	**22** / Otro sueldo
13 / Teléfono	**23** / Ingresos no previstos
14 / Gas	**24** / Rentas
15 / Ropa	**25** / Intereses
16 / Diversiones	
17 / Periódicos	
18 / Comida	

El número al lado de cada voz constituye su «código». Los números inferiores a 20 indican la cuenta de gastos, mientras que los conceptos de la 21 a la 25 indican la cuenta de ingresos. Con esta codificación la máquina puede determinar si una cantidad se ha de sumar (cuenta de ingresos, código 20) o restar (cuenta de gastos, código 10).
La introducción de los saldos deberá incluir, además de la cantidad gastada o ingresada, algunos conceptos accesorios, como:

Progresivo	3 caracteres
Fecha	6 caracteres
Causal	2 caracteres
Importe	7 caracteres
Descripción	20 caracteres
Operador	10 caracteres
—	12 caracteres (espacio disponible para ampliaciones)
	60 caracteres en total

Describamos ahora el significado de cada campo (el formato del record está en la pág. 262).

Progresivo. Indica el número progresivo del movimiento de introducción de datos. Empieza con el valor 1 (primer movimiento) y aumenta una unidad a cada introducción; de hecho, es el número del record donde serán escritos los datos introducidos, y es también la clave para releer dichos datos. Este número, para evitar errores, será suministrado directamente por la máquina: el programa memorizará el último valor introducido y, a cada nueva entrada de datos, sumará 1 a dicho valor, grabando el resultado en el campo Progresivo.

Fecha. Es la fecha del movimiento expresada en la forma: día (2 cifras), mes (2 cifras), año (2 cifras; por ejemplo, 84 para 1984).
Este campo, además de servir de recordatorio, cumple la función de «selector» en las impresiones de los balances. Efectivamente, mediante la fecha de los movimientos podemos pedir balances correspondientes a un día concreto (por ejemplo, para controlar el estado de la cuenta corriente); es decir, podemos «seleccionar» algunos de los datos presentes en el file.

Causal. Es el valor numérico al que corresponden las distintas partidas del balance. Nuestro plan de cuentas prevé códigos comprendidos entre 11 y 25; bastan, así, dos caracteres.
En la fase de introducción habrá que controlar que el valor de este campo esté entre los valores previstos (de 11 a 18 o de 21 a 25).

Importe. Es la cantidad gastada o ingresada en el movimiento; se han previsto 7 caracteres, o sea un máximo de 9.999.999.
Este límite afecta solamente a cada movimiento por separado; en la fase de elaboración de totales el límite es más alto.

ESTRUCTURA DE DATOS DEL FILE DE GESTION DEL BALANCE FAMILIAR

Record lógico: longitud 60 caracteres

El número del record coincide con el Progresivo

	1 — 3	4 — 9	10 — 11	12 — 18	19 — 38	39 — 48	49 — 60
	Progresivo	Fecha	Causal	Importe	Descripción	Operador	Espacio disponible
Longitud del campo	3	6	2	7	20	10	12

Descripción. Este campo de 20 caracteres tiene por objeto comentar el movimiento. Puede servir, por ejemplo, para detallar con una explicación la partida del plan de cuentas (Causal).

Operador. Puede ser el nombre de quien ha efectuado el pago (en la cuenta de ingresos) o de la persona que ha hecho el gasto (en la cuenta de gastos). Este campo puede servir de selector para conocer el total de los gastos efectuados por un miembro de la familia, o el total de los ingresos procedentes de una determinada fuente. Las rentas, por ejemplo, pueden ser de distintos tipos; especificando el tipo en este campo, se puede tener, mediante las oportunas selecciones, una visión inmediata de cuáles han sido los ingresos por cada tipo de renta.
El último campo está disponible para ampliaciones futuras. En la estructura de los archivos conviene dejar espacio sobrante para no correr el riesgo de no poder ampliar los datos y tener un file inutilizable en caso de que surjan nuevas necesidades.
Las características de cada campo son las siguientes (de ahora en adelante, con la abreviatura «ch» nos referiremos a «caracteres»):

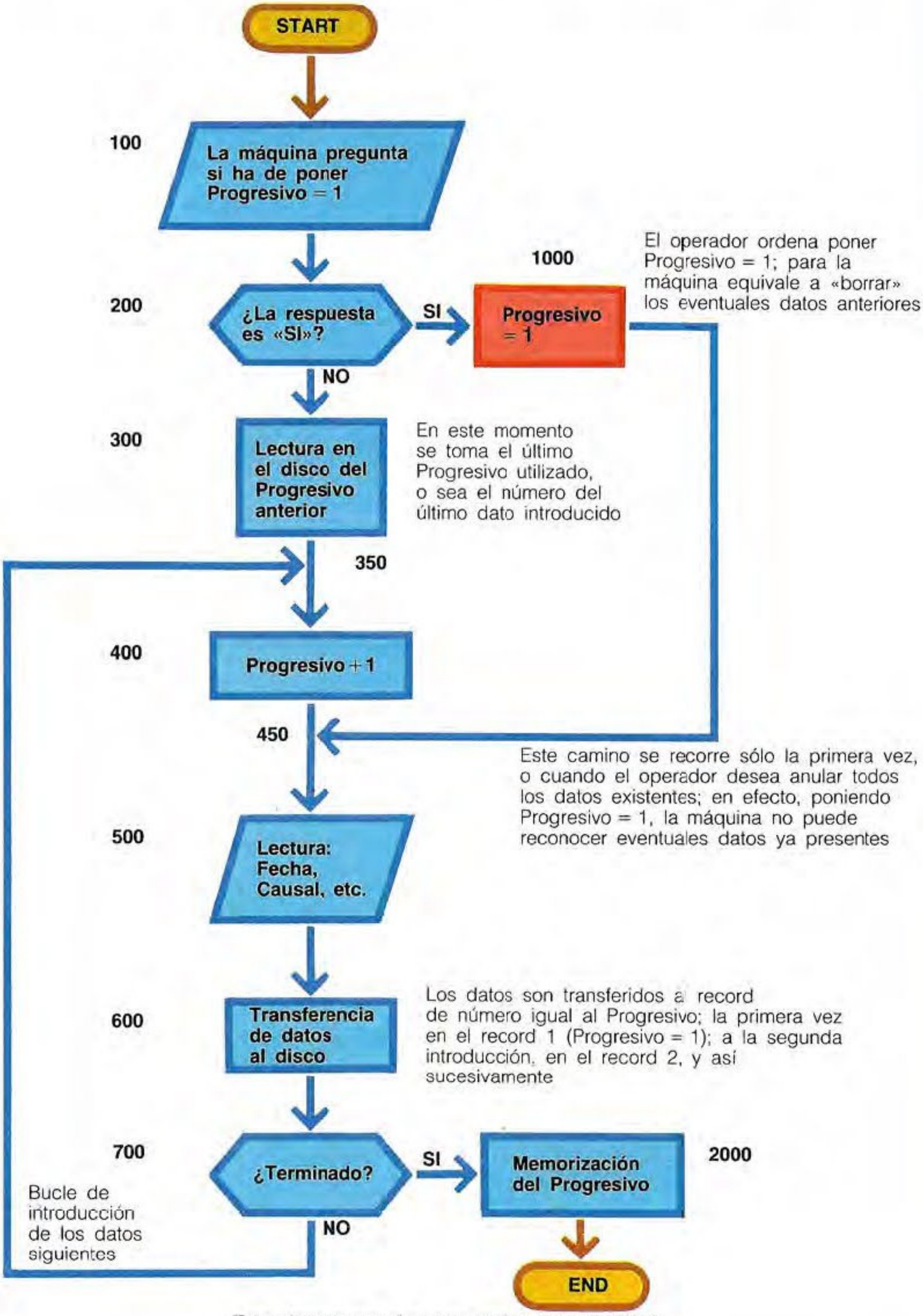
Progresivo	3 ch:	incremento automático de 1 a cada introducción
Fecha	6 ch:	control de congruencia entre día y mes (por ejemplo, no puede existir la fecha 31/02)
Causal	2 ch:	ha de hallarse entre 11 y 18, o entre 21 y 25.
Importe	7 ch:	en este campo sólo se introducen números.
Descripción	20 ch:	ningún control
Operador	10 ch:	ningún control

De esta lista se desprende la necesidad de prever dos módulos del programa, uno que realice las funciones necesarias para el automatismo del cálculo del Progresivo, y otro que controle los campos Fecha, Causal e Importe.
Antes de describir los diagramas de flujo del programa, hay que definir estos dos módulos accesorios, puesto que el primero (cálculo del Progresivo) condiciona notablemente la estructura del file.

Módulo para el cálculo del Progresivo

Con la primera introducción, y sólo entonces, se ha de poner el Progresivo = 1; cada introducción subsiguiente ha de incrementar el Progresivo en 1.
Al final de la tarea de introducción, el módulo ha de memorizar el último valor del Progresivo y retomarlo cuando el usuario reemprenda la tarea. Esta memorización se ha de efectuar necesariamente en disco (o cinta); efectivamente, si se utilizara sólo la memoria de la máquina, el valor se perdería a la primera desconexión (recordemos que las memorias electrónicas son «volátiles», es decir, pierden su contenido al faltar la alimentación). El diagrama de flujo correspondiente a las funciones mencionadas se muestra en el gráfico contiguo.
En esta lógica hay dos graves defectos que perjudican la funcionalidad del programa.

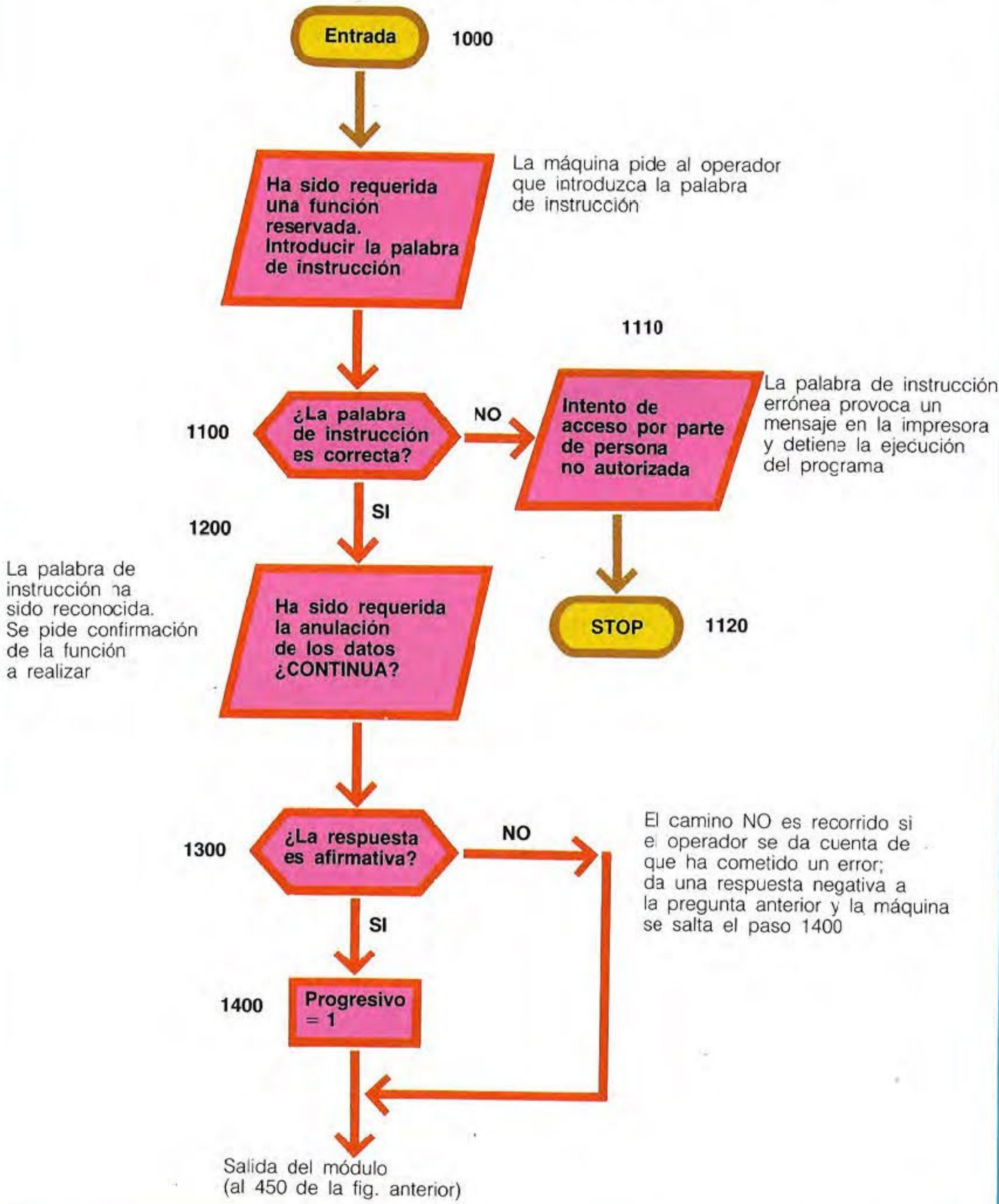
DIAGRAMA DE FLUJO DE INTRODUCCION DE DATOS

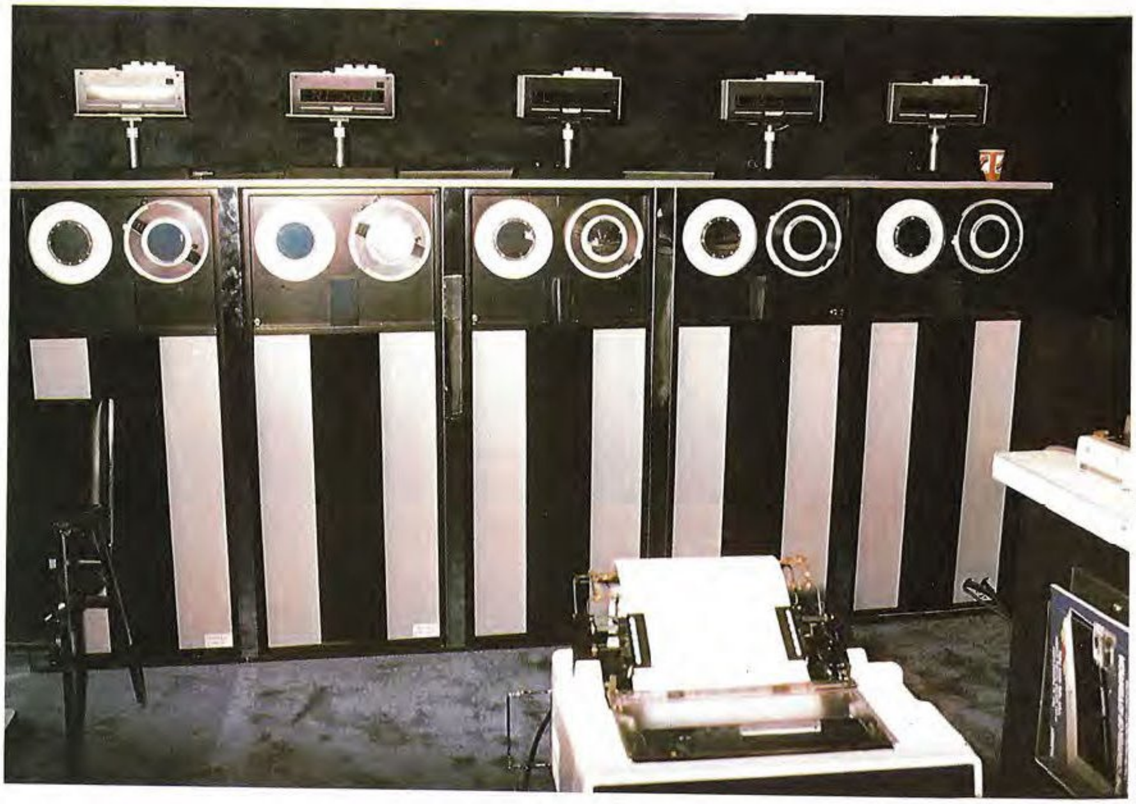


Este sistema no ofrece garantías de seguridad

El módulo con el número 1.000, que pone el Progresivo = 1, si no se usa correctamente, causa la pérdida de todos los datos introducidos con anterioridad. Efectivamente, supongamos que hemos llegado a introducir el dato número 176; la máquina recuerda este valor, puesto que lo ha memorizado en el Progresivo; el siguiente dato será memorizado en la posición

DESARROLLO DEL MODULO 1000 PARA INCLUIR LA PALABRA DE INSTRUCCION Y LA CONFIRMACION PARA CADA ACCION CAPAZ DE PROVOCAR LA PERDIDA DE DATOS





Detalle de la memoria masiva de un banco de datos.

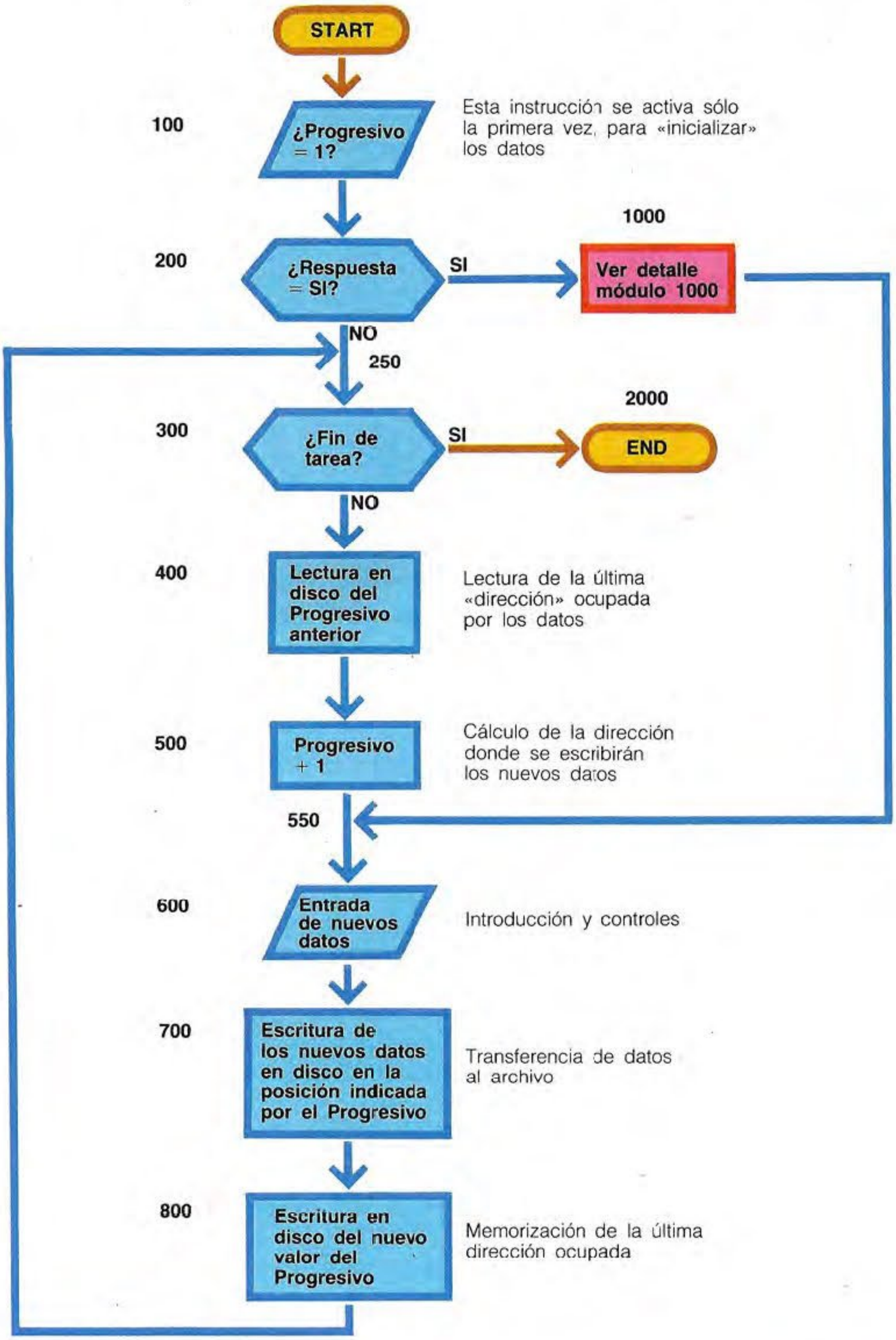
176 + 1, es decir, en una zona vacía. Por el contrario, si tras el dato 176 activamos por error el módulo 1.000, el siguiente dato será escrito en la posición 1 (el módulo 1.000 pone el Progresivo = 1). Así, el nuevo dato, que habría debido ocupar la posición 177, en realidad se superpone y por tanto «borra» lo que anteriormente se había escrito en la posición 1. Y no sólo eso, sino que entonces la máquina ya no conoce la verdadera cantidad de datos presentes, y los 176 datos anteriores serán cubiertos por las nuevas introducciones; a causa del error, estas introducciones no se sitúan detrás de la 176, sino que se escriben a partir de 1.

La única forma de minimizar el riesgo de error es poner la ejecución del módulo 1.000 bajo la condición de reconocer una «palabra instrucción» y, tras el reconocimiento, plantear una pregunta de confirmación. El método de la palabra de instrucción y repetición de la pregunta es muy frecuente en todos los programas en que un error o un uso incorrecto puede causar la pérdida de los datos. En este caso concreto, la pérdida de datos es sólo aparente o, en cualquier caso, muy limitada (basta poner nuevamente el Progresivo = 176 para recuperarlos). En otros casos, la pérdida puede ser permanente, con consecuencias desastrosas: piénsese en lo que podría suponer la destrucción de los archivos que recogen los datos económicos de grandes empresas. El diagrama de flujo del módulo 1.000, completado con los debidos controles, se muestra en el gráfico de la pág. 264. El segundo defecto se verá más claramente con un ejemplo. Supongamos que introducimos los datos y que llegamos al número 121: dicho valor no ha sido transferido aún al disco (la transferencia tiene lugar sólo al finalizar la tarea; ver diagrama de pág. 263); en este momento falla el suministro eléctrico.

Al reconectar el sistema, el valor del Progresivo ya no es 121, y todos los datos introducidos antes del corte de corriente, a pesar de estar en el archivo, ya no pueden considerarse adquiridos, pues el sistema ha perdido su «dirección».

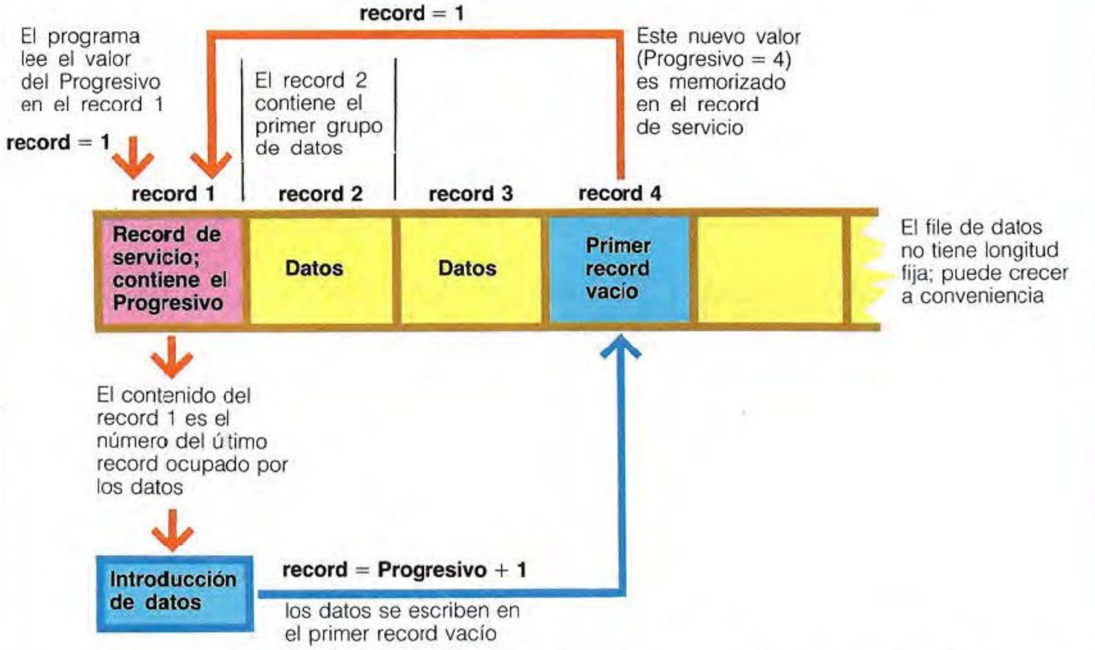
La solución de este problema es inmediata: hay que memorizar el Progresivo en cada introducción de datos, y no posponer esta operación hasta el final del programa. El diagrama definitivo y correcto se muestra en la pág. 266.

DIAGRAMA DE FLUJO DE INTRODUCCION DE DATOS MODIFICADA POR LA MEMORIZACION CONTINUA DEL PROGRESIVO

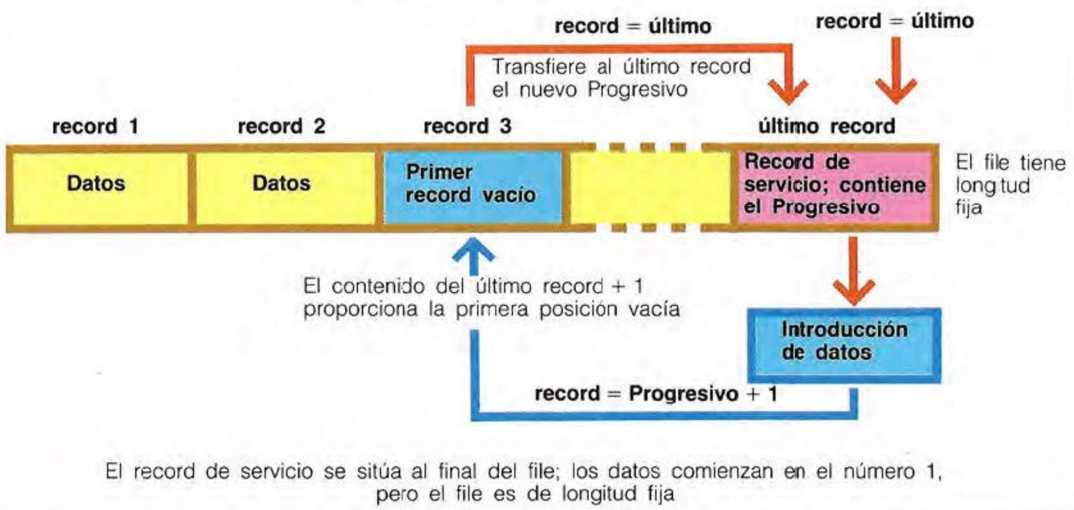


En este momento se plantea la pregunta: ¿en qué punto del disco habrá que escribir el Progresivo? Hay dos soluciones posibles. La primera consiste en utilizar un record especial del file de datos que tenga todos los campos vacíos excepto el Progresivo. Este record, aunque forma parte del file de datos, no es un verdadero dato, sino un record de servicio. El record de servicio puede situarse al comienzo del file de datos, y en él podremos leer o escribir el Progresivo. Con esta solución, el primer record útil para la grabación de datos es el número 2 y, por tanto, no existirá el dato número 1. En compensación, podremos utilizar un file largo que puede crecer durante su utilización, puesto que no es necesario establecer a priori su longitud.

ESTRUCTURA DE LOS FILES DE DATOS QUE CONTIENEN EL RECORD DE SERVICIO



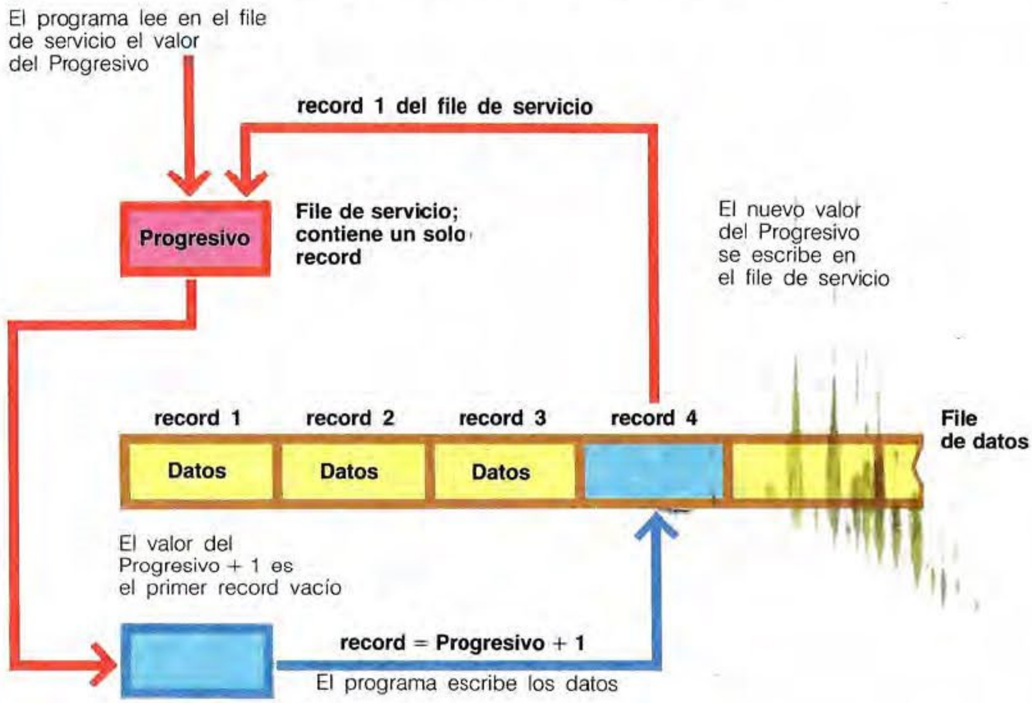
El record de servicio se sitúa en posición 1; los datos comienzan en el número 2 y la longitud del file no ha de ser fijada de antemano



El record de servicio se sitúa al final del file; los datos comienzan en el número 1, pero el file es de longitud fija

Otra posibilidad es poner el record de servicio al final del file de datos. Con este método, la numeración de los datos comenzará en el número 1 (el primer record ya no es el de servicio), pero tendremos el inconveniente de tener que establecer a priori la longitud del file de datos, puesto que la dirección del record de servicio es el último record del file. En el gráfico de la pág. 267 se esquematizan las dos situaciones con los correspondientes «recorridos» de acceso a los records. El segundo método consiste en crear un nuevo file que contenga un solo record, en el que se memoriza el Progresivo. El método permite iniciar la numeración de los datos por el 1 conservando las ventajas del file de longitud no preestablecida. El único inconveniente es que hay que gestionar dos files simultáneamente: el file de datos y el file de servicio.
En el gráfico inferior se muestra esta estructura y la lógica de utilización de la misma.

ESTRUCTURA Y FLUJOS DE INFORMACION CON LA GESTION DE LOS PUNTEROS EN FILE APARTE



Cualquiera que sea el método elegido, las funciones de lectura y escritura del Progresivo se realizan en los módulos 400 y 800 del diagrama

de flujo general de la pág. 266. Estos módulos se desarrollarán según el método elegido. En el gráfico siguiente se ve el módulo de lectura en caso de utilizar el file de servicio denominado DIR; el módulo de escritura (800) se desarrolla de forma idéntica (sustituyendo la lectura por escritura, también en el file DIR record 1).

LECTURA DEL VALOR DEL PROGRESIVO EN EL FILE DIR

Módulo: 400

Objetivo: lectura en el file de servicio del campo Progresivo

Nombre del file: DIR

Longitud: 1 record (69 bytes)

Posición del campo Progresivo: 1-3

Salida: valor numérico del Progresivo



Salida hacia la instrucción 500

La escritura (módulo 800) tiene el mismo desarrollo

Módulo de control de introducciones

Como hemos visto, en la fase de introducción de datos hay que comprobar la exactitud formal de los campos Fecha, Causal e Importe.

Esta función ha de realizarse en el interior del módulo de introducción de datos (600, en el gráfico de la pág. 266), de forma que no sea posible la salida del módulo sino con los datos exactos. En la pág. 270 se desarrolla el diagrama de flujo detallado del módulo. En el diagrama se muestra el módulo 620 de control de la fecha. Puesto que se trata de una función de uso generalizado, se desarrollará con detalle más adelante; por ahora, admitamos que disponemos de un módulo capaz de controlar la congruencia entre día y mes y de responder con un valor de error. Si dicho error es cero, la fecha es

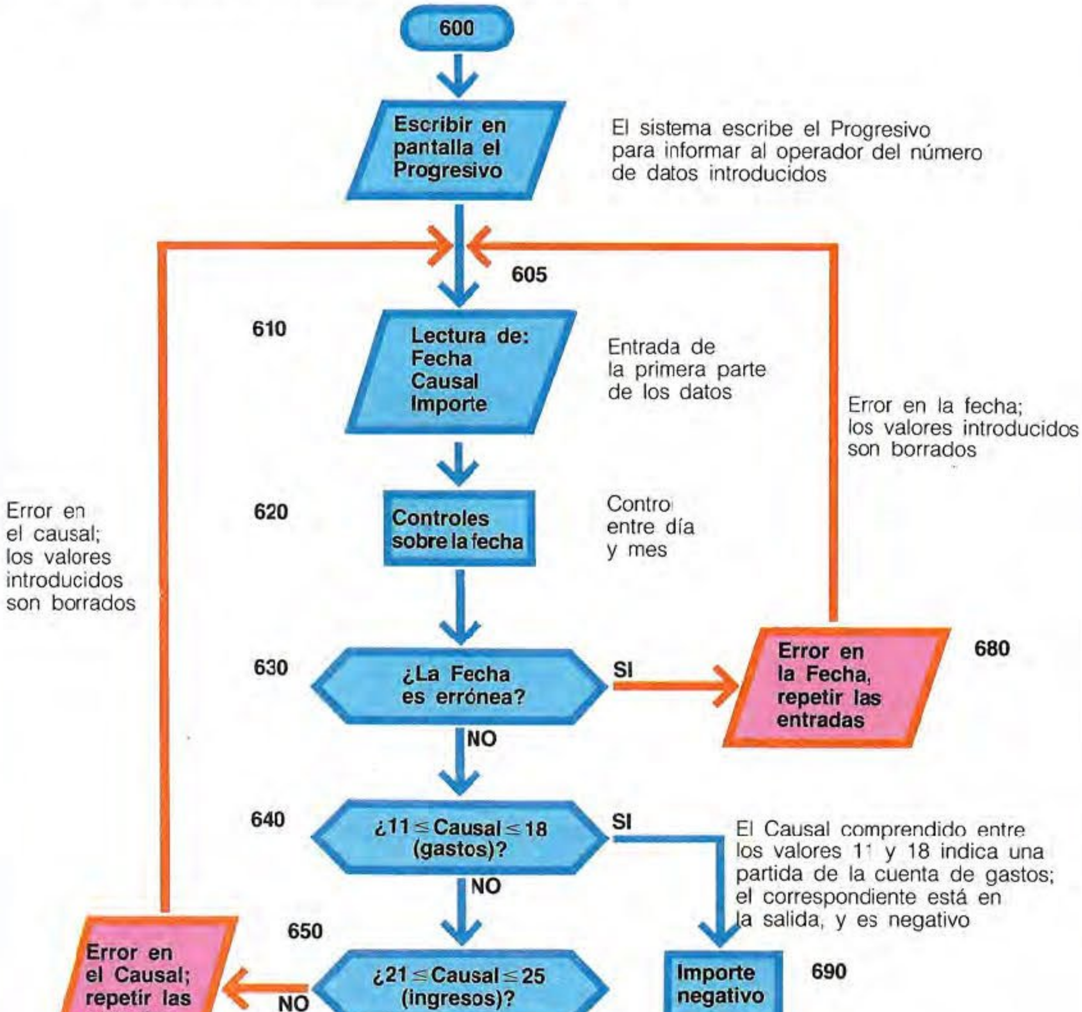
DIAGRAMA DE FLUJO DETALLADO DEL MODULO DE ENTRADA Y CONTROL DE DATOS

Módulo: 600

Objetivo: lectura por teclado de nuevos datos y control formal

Entrada: Progresivo

Salida: campos de datos listos para transferencia a disco



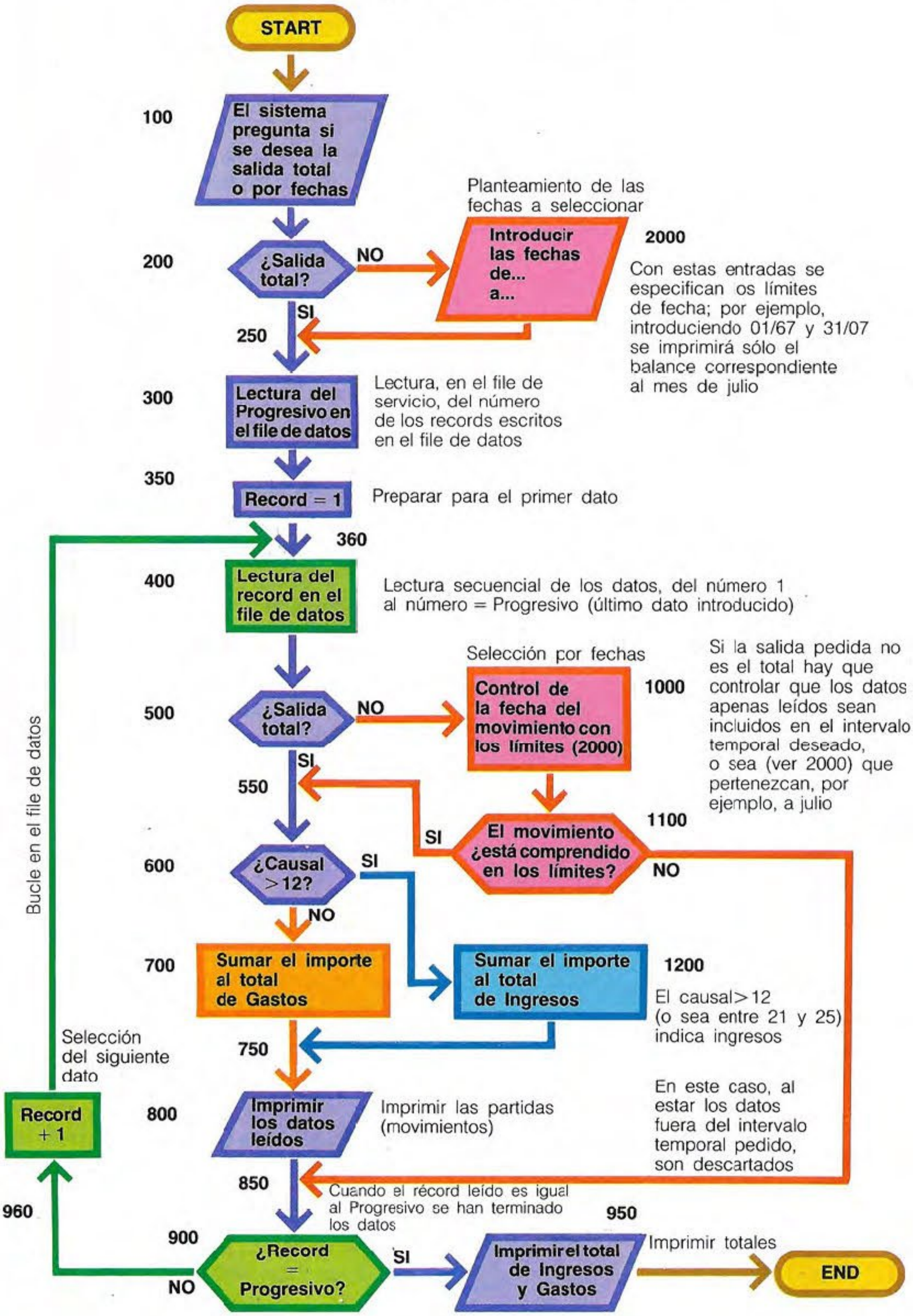
correcta; de lo contrario es errónea. El valor numérico del error indica qué clase de equivocación se ha cometido con la fecha.
Este módulo puede servir también para dar una «caducidad» a los programas. Si, por ejemplo, imponemos la condición de que el año no pueda ser superior a 1995, en 1996 el programa no aceptará más entradas. Este valor límite no puede ser modificado por el usuario si se le suministra la versión del programa compilado; sólo quien posea el programa fuente puede efectuar correcciones. Naturalmente, al final de las correcciones, el programa tendrá que ser recompilado, obteniéndose así la nueva versión.
Al usuario le queda la alternativa de suministrar al sistema una fecha ficticia, por ejemplo, seguir introduciendo 1995 en lugar de 1996; pero dicha solución sólo puede emplearse en programas de uso personal; en un programa de uso amplio, por ejemplo para documentos fiscales, no pueden aparecer fechas falsas.
Esta técnica de caducidad permite alquilar los programas con la seguridad de que el usuario pagará el alquiler, de lo contrario, al caducar en fecha establecida por el programador, la falta de actualización del bloque de la fecha hará que el programa deje de funcionar.
El módulo 600 permite la introducción de los datos. Ahora hay que completar el procedimiento con el desarrollo de los cálculos.
La función que ha de realizar este nuevo programa consiste sólo en leer todo el file de datos y sumar por separado los importes de las cuentas de gastos e ingresos. De esta manera tendremos una visión completa de todos los movimientos efectuados y el saldo global, es decir, la diferencia entre el total de ingresos y el de gastos. A esta función de totalización hay que añadir la posibilidad de efectuar un balance entre dos fechas determinadas. Así, podremos solicitar la situación global (desde el comienzo de los datos hasta el final) y la de un determinado período. El diagrama de flujo de este programa se muestra en el gráfico de la pág. 272.
Los programas «accesorios», es decir, los que permiten efectuar selecciones concretas (Causal, Descripción, Operador), se describirán más adelante de forma general. Los detalles de cada uno de los módulos que aparecen en la pág. 272 se ilustrarán junto con el procedimiento de escritura del programa, tras la descripción de las correspondientes instrucciones Basic.

Resumiendo:
- **en la pág. 263 se muestran la estructura del record de datos y las longitudes de los campos;**
- **en la pág. 266 se muestra el diagrama del programa completo de introducción y control de datos;**
- **la función de iniciación la realiza el módulo 1.000, cuyo diagrama se muestra en la pág. 264;**
- **la gestión del Progresivo de datos se logra con un file auxiliar de servicio (ver gráficos de páginas 268 y 269);**
- **las funciones de introducción y control de datos pueden verse en la pág. 270. Traduciendo por orden los diagramas de las págs. 266, 269, 270 y 264, tendremos listo el programa.**

La búsqueda de datos en los archivos

Una vez establecido el archivo (fase de introducción) hay que tenerlo actualizado, es decir, aportar a los datos todas las variaciones necesarias (fase de actualización). Para actualizar un record hay que leerlo y, según el contenido, efectuar los añadidos y modificaciones.
La operación principal del procedimiento de actualización es la búsqueda de los datos. En la fase de lectura del record, el usuario conoce la clave y ha de extraer todos los datos correspondientes a ella para decidir cuáles modificar.
Durante la utilización normal del archivo, la necesidad más frecuente es la de extraer los datos según una lógica determinada; por ejemplo, en un archivo de direcciones, todas las direcciones de una determinada ciudad, o en un archivo del personal, todos los pertenecientes a una determinada categoría.
La realización de listados requiere la extracción de parte de los datos contenidos en el archivo. Por lo tanto, hacen falta programas de búsqueda que encuentren todos los records cuyos datos tienen las características requeridas.

DIAGRAMA DE FLUJO DEL PROGRAMA DE CALCULO E IMPRESION DEL BALANCE



Los principales métodos de búsqueda son dos: la **búsqueda en serie** y la **búsqueda binaria** (o **dicotómica**).

Búsqueda en serie

La búsqueda en serie de un dato se efectúa leyendo uno tras otro todos los datos contenidos en el archivo y comparándolos con el valor de la clave de acceso; si hay coincidencia, el dato ha sido hallado y puede ser enviado a impresión; de lo contrario, la búsqueda continúa hasta que se terminen los records.

El número promedio de lecturas necesarias para encontrar un dato viene dado por la siguiente relación sencilla:

$$\textbf{número promedio de lecturas} = \frac{\textbf{longitud del file} + 1}{2}$$

Así, en un file con 1.000 datos tendremos que leer, por término medio, unos 500 antes de encontrar el que nos interesa.

Como ya se ha dicho, el dato a seleccionar no tiene por qué estar una sola vez en el archivo. En general, podría estar contenido en varios records y, en este caso, la búsqueda tendría que continuar hasta que se terminasen los datos. La única forma de acelerar el proceso consiste en ordenar los datos antes de activar el programa de búsqueda.

Dos fases del trabajo que se realiza en un banco de datos. La planificación y estructuración de la información es el planteo esencial para una utilización óptima de la memoria.

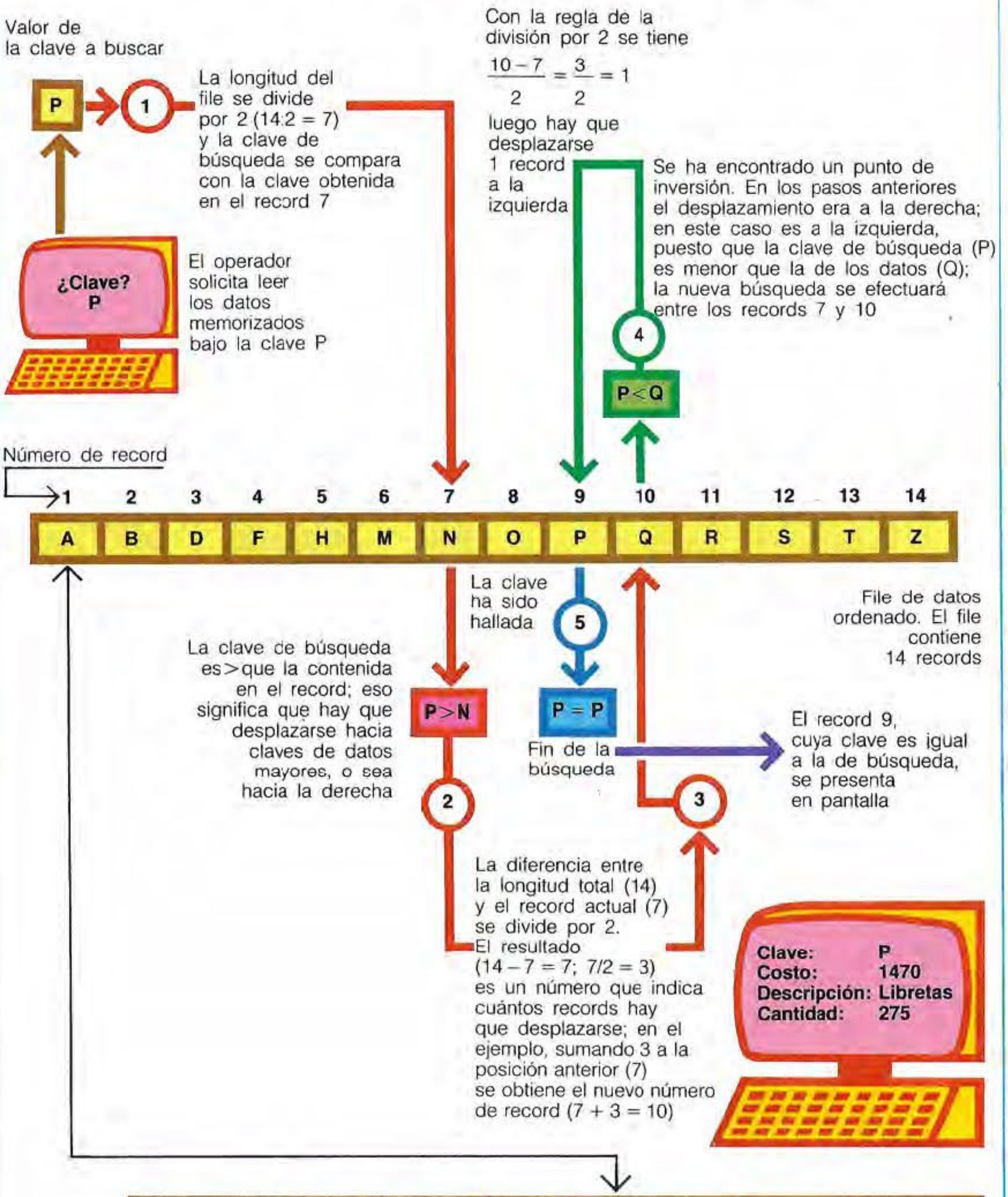


Marka



Marka

ESQUEMA LOGICO DE UNA BUSQUEDA DICOTOMICA



	Clave	Costo	Descripción	Existencias
Contenido	**A**	**12.560**	**Plumas estilográficas**	**106**
Número de caracteres	**1**	**6**	**20**	**3**

Longitud del record = 1 + 6 + 20 + 3 = 30 bytes

Debido a los largos tiempos de elaboración, la búsqueda en serie se utiliza sólo para archivos pequeños, para los que el empleo de otros métodos no ahorra tiempo importante.

Búsqueda binaria

La búsqueda binaria (o dicotómica) puede efectuarse sólo en files ordenados. Consiste en leer los records a «saltos» que dividen los datos por la mitad, comparando la clave de los datos con la de búsqueda y, en base al resultado de la comparación, se decide la dirección del siguiente salto (a la derecha del punto actual si la clave de búsqueda es mayor que la del dato, a la izquierda en caso contrario). Aclaremos el concepto con un ejemplo (ver gráfico contiguo).
Un almacén de artículos de papelería está estructurado en un file con una longitud de record de 30 bytes, divididos en estos campos:

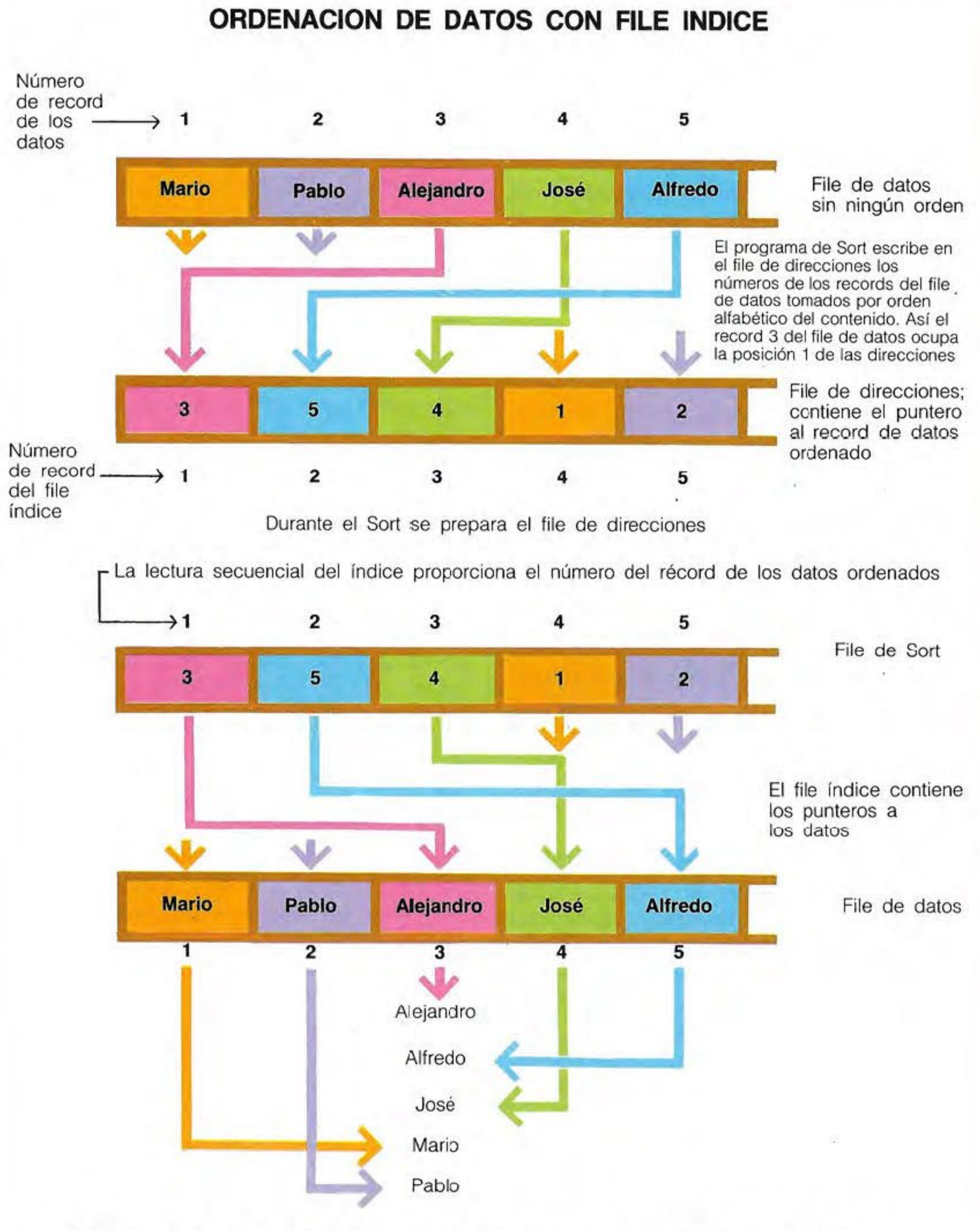
- Campo clave: alfabético, 1 carácter.
- Costo del artículo: numérico,* 6 caracteres.
- Descripción del artículo: alfabético, 20 caracteres.
- Existencias: numérico*, 3 caracteres.

El usuario desea conocer el contenido del record cuya clave es la letra P e inicia la búsqueda dicotómica.
En el primer paso (punto **1**), se divide la longitud del file por 2; en nuestro ejemplo, 14/2 = 7. El contenido del campo clave del record 7 (punto **2**) se compara con la clave de búsqueda (P). El dato está asociado a la letra N (punto **2**); eso significa que la clave P (P>N) ha de encontrarse a la derecha del record examinado, es decir, comprendida entre el record 7 y el 14 (fin del file). La zona a examinar (del record 7 al 14) se divide de nuevo por 2; el valor que se obtiene (3) es el desplazamiento, en número de records, a efectuar hacia la derecha.
El record leído anteriormente era el número 7; ahora hay que desplazarse tres records hacia la derecha, es decir, hay que leer el record 7 + 3 = 10 (punto **3**). En este record hay una clave mayor que P (punto **4**); por lo tanto, la clave buscada debe hallarse a la izquierda de este record. Antes se había determinado que la clave debía hallarse a la derecha del record 7, por lo que ha de estar entre el record 7 y el 10. De nuevo el número de records comprendidos en la zona a examinar (10-7) se divide por 2 y el resultado (entero) se toma como desplazamiento a efectuar a la izquierda del punto actual. El record que estamos examinando es el número 10, por lo que el desplazamiento de 1 a la izquierda nos lleva al record 9. Este grupo de datos contiene la clave P (punto **5**); la búsqueda ha terminado, y los datos del record 9 aparecen en la pantalla.
Con este tipo de búsqueda, entre los 14 datos contenidos en el file se han efectuado sólo tres lecturas (records 7, 10 y 9). Si hubiéramos adoptado la búsqueda en serie, habríamos tenido que efectar nueve lecturas (los nueve primeros records, hasta encontrar la clave P). Por tanto, las ventajas de la búsqueda binaria sobre la búsqueda en serie son evidentes. Sin embargo, hay que tener en cuenta algunas cosas. La búsqueda binaria sólo se puede aplicar a files ordenados; si el file de datos no tiene orden alguno, antes de iniciar la búsqueda hay que reescribirlo ordenadamente. En este caso, además del tiempo necesario para la búsqueda, hay que considerar el que se emplea en ordenar el file. Para los files con un número reducido de records, esta solución puede no ser adecuada (en términos de tiempo y complicaciones del programa) respecto a la búsqueda en serie.

Ordenación de los datos

Uno de los principales problemas en la gestión de archivos es la ordenación de los datos, condición básica para aplicar técnicas de elaboración (como la búsqueda dicotómica) o para obtener listados fáciles de consultar.
Si en un archivo, aunque no sea muy grande, no hay posibilidad de obtener un listado de los datos ordenados, es poco menos que imposible encontrar los elementos requeridos. Sería como tener una guía telefónica que, en vez de estar ordenada alfabéticamente, contuviera los nombres de los abonados por el orden de las fechas en que contrataron el servicio telefónico. De esta forma, el señor Pérez, antiguo abonado, aparecería antes que el señor López, y sería imposible encontrar un determinado número de teléfono, puesto que no podríamos aplicar ninguna

* En muchos ordenadores personales los valores numéricos se memorizan en disco como caracteres; por ejemplo, el valor numérico 579 se memoriza en tres bytes que contienen, por orden, las cifras 5, 7 y 9 codificadas como caracteres, o sea con los valores hexadecimales 35 (=5), 37 (=7) y 39 (=9) (ver código ASCII). En otros ordenadores, los valores numéricos pueden ser transferidos al disco como tales; en este caso, la ocupación de bytes es menor. Este tema será ampliado más adelante.



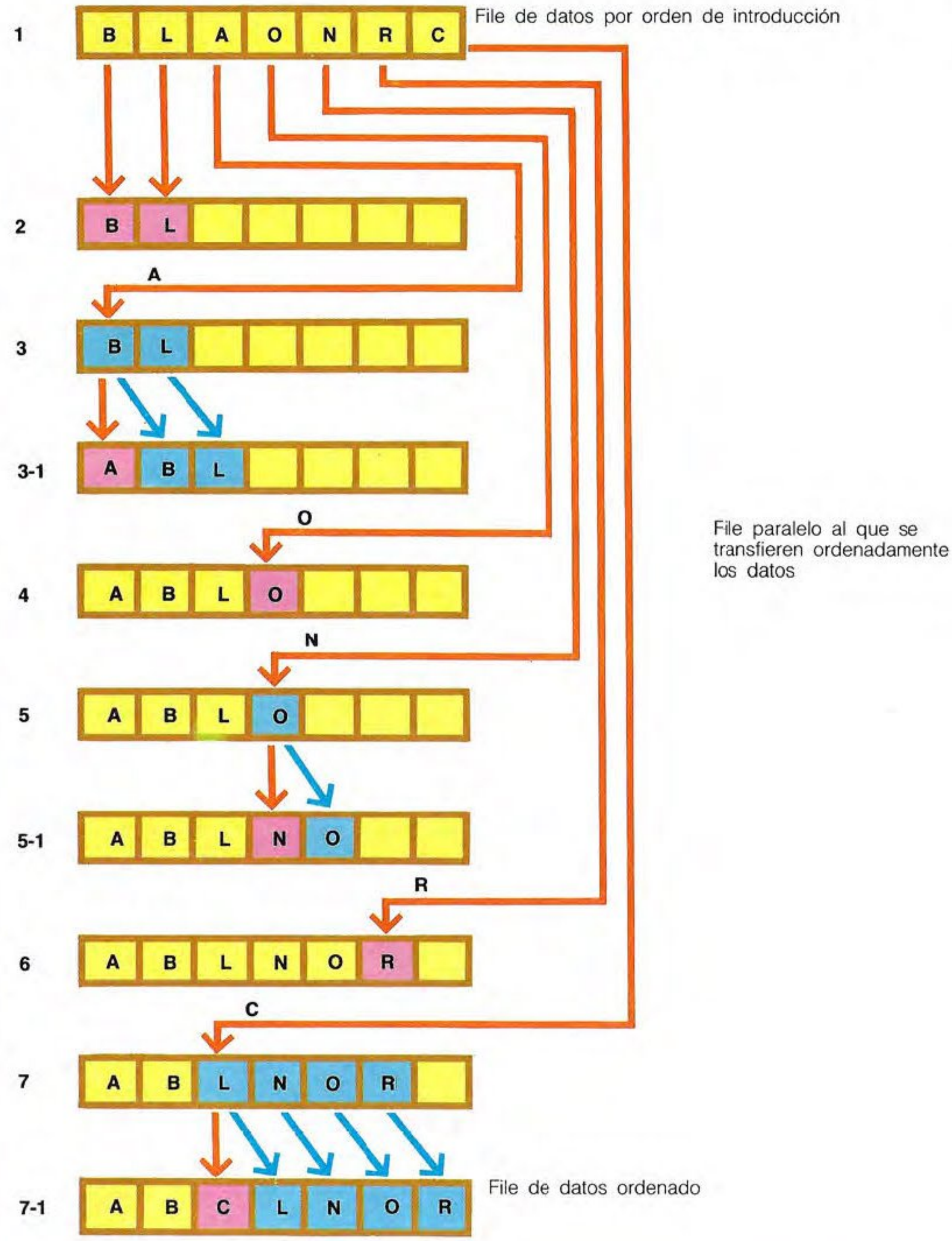
Para obtener una impresión ordenada del file de datos, hay que leer secuencialmente el file índice, usando su contenido como puntero al file de datos

lógica de búsqueda en tiempo real. Análogamente, si en un archivo sólo se hubiera previsto la impresión por orden de introducción, no podríamos utilizar los datos que contiene.

Por estos motivos, en todos los procedimientos

de gestión de archivos tienen que haber programas especiales capaces de ordenar los datos. Estos programas reciben la denominación genérica de **Sort.**

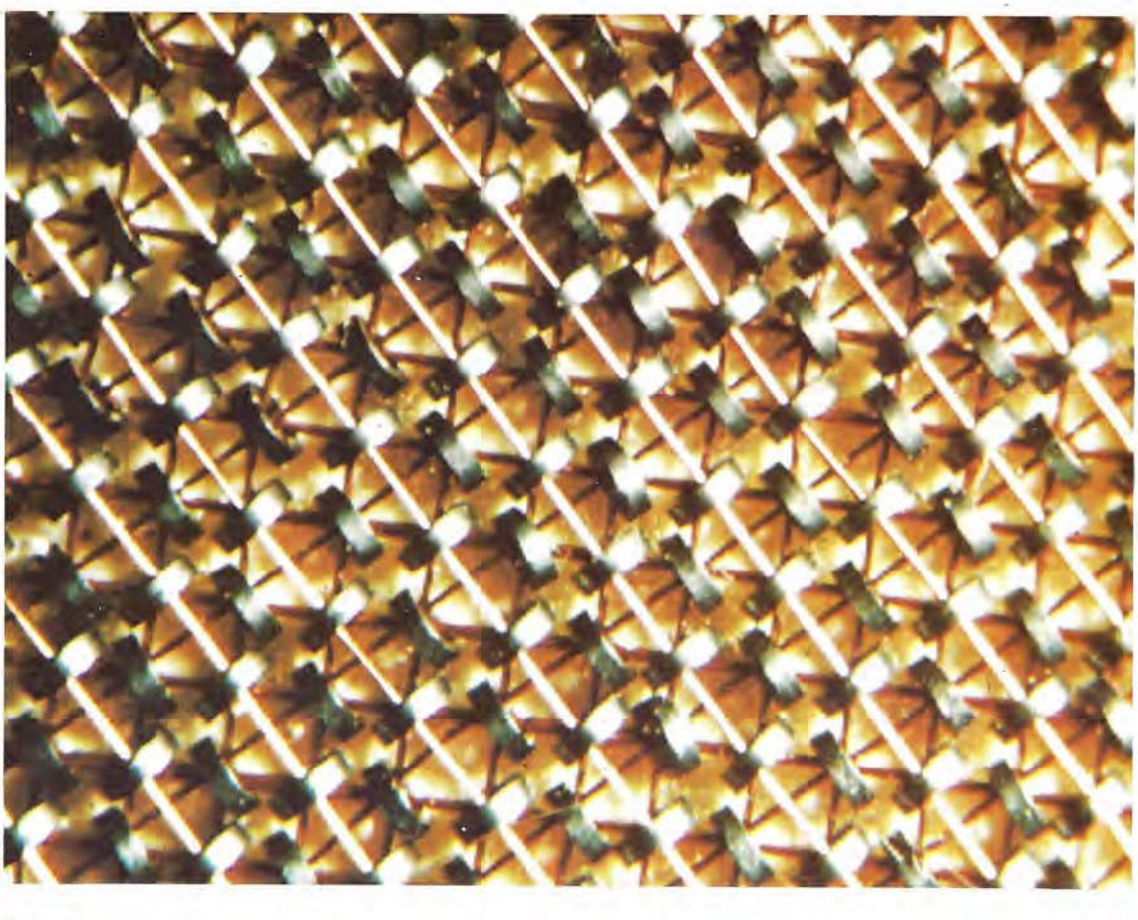
ORDENACION DE DATOS CON CONSTRUCCION DE UN FILE PARALELO



Dada la importancia de la función realizada por los programas de Sort, están incluidos en casi todos los sistemas operativos y, por razones de velocidad, suelen estar escritos en Assembler. En estos Sort de «utilidad», es decir, de uso generalizado, el usuario ha de suministrar el nombre del file de datos, la estructura del record (en cuántos campos está dividido y de qué longitud) y la clave con que desea ordenar el file.
Los programas de Sort pueden operar según dos técnicas principales. La primera consiste en copiar el file de datos ordenándolo en otra zona del disco; se obtiene así un nuevo file, con distinto nombre, que contiene los datos ordenados. La segunda consiste en elaborar un file de direcciones que contiene las direcciones ordenadas de los datos del primer file.
Con este último método, si se desea leer los datos de forma ordenada, hay que leer primero el file de direcciones y luego el record de los datos apuntados por la dirección. La lógica de este método se muestra en el gráfico de la pág. 276. Este segundo método, además de ser más rápido (no ha de copiar todo el file sino sólo escribir direcciones), ocupa menos espacio en el disco porque no crea copias de datos ya existentes.
Los programas de Sort están incluidos en todas las máquinas de categoría mini, pero pueden faltar en los ordenadores personales y microordenadores. Para estas máquinas, el propio programador ha de escribir el programa de Sort.
Este tipo de programas plantea algunas dificultades que sólo se superan procediendo con mucho método, elaborando una documentación clara y diagramas de flujo detallados. Para mostrar la metodología a seguir, desarrollaremos un ejemplo de ordenamiento de un file con uso de file paralelo (ver gráfico de pág. 277).

1 / Una vez leído el primer record de datos (contiene B), se transfiere al primer record del file de salida (este file «paralelo» contendrá los datos ordenados).

Los modernos bancos de datos requieren ingentes capacidades de memoria, muy superiores a las de los primeros dispositivos magnéticos de núcleos de ferrita.



Marka

TEST 7

1 / ¿Cuál de estas afirmaciones es cierta para un file secuencial?
a: se puede acceder a los records en un orden cualquiera;
b: los datos se memorizan por orden creciente del campo clave;
c: los records se escriben y leen secuencialmente.

2 / Para un file con índices ¿cuáles de estas afirmaciones son ciertas?
a: el índice es un campo especial que contiene datos sobre los records;
b: permite efectuar la ordenación de los datos rápidamente;
c: un file de este tipo puede tener más de un índice.

3 / ¿En qué consiste un Sort?
a: extracción de los datos por orden;
b: búsqueda, en los datos de un file, de un determinado valor;
c: duplicación del file.

4 / Enumerar las principales diferencias entre búsqueda en serie y dicotómica.

5 / Un archivo de personal ha de contener:
nombre, apellido, cualificación, nivel, antigüedad, sueldo base, faltas de asistencia. Definir el formato del record con una hipótesis sobre distintas longitudes de los campos.

6 / Definir la estructura del record de un file para la gestión de un almacén. Los campos a prever son los siguientes: artículo, existencias, reserva mínima, fecha del último pedido, nombre y apellido del proveedor.

7 / En la estructura de la cuestión 6 se puede ahorrar espacio codificando algunos campos. Codifíquese la fecha de pedido (día, mes, año) y el nombre del proveedor (suponiendo que los proveedores sean cien como máximo).

Las soluciones, en las págs. 286 y 287.

2 / El segundo record de datos (contiene L) se transfiere al segundo record del file de salida, puesto que su contenido es mayor que el del primero (L>B).

3 / El record 3 (datos) contiene la letra A, y es por tanto menor que todos los datos ya contenidos en el file de salida (B y L), por lo que ha de situarse en la posición 1.

3-1 / Puesto que la posición 1 ya está ocupada (por la letra B, escrita en el paso **1**), hay que dejar libre esta posición desplazando todo el contenido del file de salida un lugar a la derecha. La letra L, que ocupaba el record 2, pasa al 3, y la B al 2. Así se deja libre el record 1 para la letra A.

4 / La letra O (record 4 de los datos) se sitúa en la última posición, puesto que es mayor que todo el contenido del file.

5 / En este caso, el nuevo record a ordenar (N) ha de ser situado antes que O. Por lo tanto, O se desplaza una posición y su lugar es ocupado por el nuevo dato (paso **5-1**).

6 / La letra R (record de datos número 6) se pone al final, puesto que es mayor que los demás datos del file de salida.

7 / Para la letra C (record 7 de los datos) hay que correr los últimos cuatro records del file de salida (L, N, O y R), dejando así un lugar vacío entre B y L.

7-1 / Aquí se aloja el último record de datos (C) y el file de salida está completo.

Bancos de datos especializados

En un mundo en continua transformación, la demanda de información por parte de entidades económicas y científicas, privadas y públicas, es cada vez más apremiante. No es posible mantener contactos actualizados con toda la literatura especializada publicada en el mundo, tanto por razones de tiempo como de costo.

De esta necesidad ha nacido el mercado de la información. Los realizadores de bancos o bases de datos recogen, seleccionan y memorizan lo más esencial de una gama de temas.

Algunos bancos de datos están sumamente especializados y cubren la literatura completa de un sector concreto. Otros –y actualmente son mayoría– son multidisciplinarios. Igualmente importantes son los bancos dedicados a la recopilación de datos estadísticos, demográficos y comerciales, que están teniendo una utilidad creciente en función directa de la capacidad del proveedor de datos para garantizar una rápida actualización de sus archivos. El documentalista que trabaja en un pequeño centro puede consultar sólo algunas de las publicaciones especializadas que lo conciernen, mientras que del banco de datos puede obtener rápidamente una lista de referencias bibliográficas actualizadas, para localizar cualquier información útil.

Hay, además, bancos de datos especializados en la difusión de patentes. De este modo es posible, incluso para una empresa modesta, comprobar si en su campo de interés se producen amenazas u oportunidades bajo la forma de nuevos productos o procedimientos.

De forma análoga se hace posible una rápida actualización en campos como la medicina, el comercio internacional, el derecho, etc.

De especial interés son las perspectivas que la industria de la información ofrece a un país como Italia, caracterizado por una estructura productiva de transformación sometida a continuas presiones del exterior, tanto por lo que respecta al mercado de los factores de producción (nuevas tecnologías de producción) como al de los productos acabados. Un sistema de este tipo debería orientar todos sus recursos hacia la innovación tecnológica, organizativa y comercial, con objeto de mantener, y a ser posible mejorar, las posiciones adquiridas. Las primeras empresas que operan en el sector de la información on-line *se han desarrollado en Estados Unidos, donde sistemas muy avanzados de elaboración de datos cuentan con la más moderna estructura de telecomunicaciones del mundo. Posteriormente, al afianzarse estos servicios y aumentar el número de usuarios potenciales, los productores estadounidenses han ofrecido sus archivos también a otros países. Mientras tanto, e inspirándose en el modelo americano, han surgido productores de información en todos los países tecnológicamente desarrollados. Algunos de ellos ofrecen servicios análogos a los estadounidenses; otros, con carácter nacional, suministran datos originales.*

Las Administraciones postales y telefónicas de los países miembros de la CEE, para hacer que este patrimonio se desarrollara según un modelo europeo y para adquirir de forma autónoma las técnicas necesarias, han creado la primera red europea, EURONET, mediante un acuerdo que, haciendo efectiva en el campo de las comunicaciones esa apertura de fronteras no siempre realizada para las mercancías, pone en libre competencia entre sí a los productores europeos de bancos de datos y permite a los usuarios un acceso indiscriminado.

En funcionamiento desde 1980, EURONET ofrece un soporte de telecomunicaciones para los bancos de datos de la Comunidad, reunidos bajo la sigla DIANE (Direct Information Access Network in Europe). La peculiaridad de esta red estriba en su internacionalismo.

El usuario que desea consultar un archivo con sede en cualquiera de los países miembros, ha de suscribir un contrato de usufructo con EURONET y disponer de un terminal conectable con la red telefónica. Desde ese momento la conexión con la red es gestionada automáticamente por los ordenadores que constituyen los «nudos» de la red.

Para el usuario italiano basta con telefonear al nudo de Roma o Milán y, sin más cargo que el de la eventual teleselección nacional, puede conectar con bancos de datos realizados en Gran Bretaña, Holanda, Alemania, Francia, Bélgica y Luxemburgo. No hay problemas de distancia, y el único costo a asumir por el usuario es el de la consulta. Hay que señalar, además, que las tarifas de EURONET son en muchos casos inferiores a las estadounidenses.

Para los bancos de datos de Estados Unidos, el procedimiento es apenas más complicado.

Para obtener el puente telefónico entre Italia y una de las numerosas networks *estadounidenses, hay que estipular un contrato con Italcable*

que importa unas 30.000 liras trimestrales; a esto hay que añadir el contrato con la o las redes norteamericanas, que, como gran parte de los bancos de datos de EURONET, sólo factura por servicio efectuado, sin cobrar mínimos obligatorios. Los archivos disponibles en las redes de telecomunicaciones están marcadamente diferenciados por contenidos y funciones. Hay una división fundamental, en función del contenido, entre «bancos de datos» y «bases de datos».
Las bases de datos son archivos en donde se memorizan, de forma consultable mediante terminal, las denominadas publicaciones secundarias, entendiendo por tales las recensiones, de forma unificada y sintética, de revistas, monografías, libros, conferencias, etc., destinadas a suministrar, a una demanda muy diversificada, un medio eficaz para la localización de documentos útiles para una consulta específica.
El productor, que normalmente es una empresa que suministra los resúmenes de revistas o publicaciones ya sobre papel, puede ser una sociedad ligada a un determinado sector productivo, y en tal caso seleccionará los escritos de carácter más especializado, o bien un instituto universitario, o una empresa privada que opera en el campo de la elaboración de datos; en ese caso, el contenido de los archivos ofrecidos es, normalmente, multidisciplinario.
Por cada artículo, cita o monografía recensionada, se indica: el título, en inglés y en lengua original; el autor; la fuente (revista, monografía, artículo), con los elementos necesarios para identificar el original; una lista de descriptores o palabras clave, pertenecientes al texto, que sirven para evaluar la pertinencia de la referencia y, sobre todo, para focalizar el tema de la búsqueda; el resumen, que da una idea sintética pero suficientemente completa del documento.
El producto final de un procedimiento de extracción en estos archivos será un conjunto de todas las referencias bibliográficas que el documentalista ha podido separar, combinando entre sí un cierto número de elementos que caracterizan unívocamente el tema solicitado.
Este servicio es especialmente útil en el caso de usuarios que residen lejos de la fuente de información principal, y que tienen, por tanto, dificultades para conseguir las publicaciones que necesitan para realizar su trabajo, pues de este modo pueden utilizar un instrumento que les permite seleccionar sólo los textos realmente útiles. Esta situación se puede dar cuando el lugar de residencia no está bien abastecido de bibliotecas, o bien si se buscan publicaciones con carácter ocasional, que no figuran entre las revistas utilizadas periódicamente. La consulta de una base de datos requiere un trabajo pre-

Los bancos de datos demuestran su utilidad sobre todo en el campo científico. En la foto, el sistema de adquisición y elaboración de datos del Observatorio Radioastronómico de Greenbank (Virginia, USA).



Marka

paratorio previo. Se seleccionan los archivos que se prevén más útiles y pertinentes a efectos de la consulta, eligiendo los especializados y aquellos de entre los multidisciplinarios de los que se espera obtener resultados satisfactorios.
En el planteamiento de la estrategia de consulta se ha de tener en cuenta el tipo de archivo consultado. Si se prevé que se va a obtener un número de referencias excesivo, a causa de un planteamiento demasiado general de la estrategia, se puede restringir el campo de búsqueda definiendo parámetros más precisos, como la fecha de actualización, el tipo de fuente, etc.
Las órdenes que permiten la definición de la búsqueda están constituidas por la secuencia, ordenada lógicamente, de las palabras que el documentalista espera encontrar en el título o en el texto y que considera específicas del tema buscado. A título de ejemplo, veamos una breve estrategia con las órdenes de extracción y la impresión de una de las referencias bibliográficas extraídas. La búsqueda se efectúa en la base de datos PAPERCHEM, producida por The Institute of Paper Chemistry, *accesible en la red estadounidense DARDO vía ITALCABLE.*
En el ejemplo se utiliza uno de los operadores que PAPERCHEM reconoce para individualizar documentos de un tipo concreto: el prefijo CT (conference title).
Se busca el informe de una conferencia titulada Energy from Biomass conference. *La consulta se efectúa con el conjunto de órdenes:*

SELECT CT = ENERGY (1W) CT = BIOMASS

El significado de la consulta es: Seleccionar, en el campo de los títulos de conferencias, el término ENERGY; en el mismo campo hay que encontrar el término BIOMASS; el operador (1W) especifica que las dos palabras han de estar en el orden indicado (BIOMASS, en el título, ha de seguir a la palabra ENERGY), y que puede haber una tercera palabra uniendo ambos términos.

? SELECT CT = ENERGY (1W) CT = BIOMASS
33 83 CT = ENERGY (1W) CT = BIOMASS

De la impresión de la referencia extraída se deduce:
1. el título del informe; 2. el informante; 3. el título de la conferencia (es el campo en que se ha efectuado la búsqueda); 4. las páginas, en el documento original, que contienen el informe específico; 5. la fecha de la conferencia; 6. el número total de informes; 7. la clasificación del documento; 8. el idioma original; 9. el resumen; 10. los descriptores.
Los bancos de datos se distinguen por dos características fundamentales. Ante todo, la información obtenida con la consulta coincide con el dato final buscado. No incluye, por tanto, ningún tipo de publicación secundaria, sino que está constituida básicamente por copias de documentos originales consultables mediante terminal. Su otra característica es la naturaleza del contenido, que es numérico, en forma de series históricas y tablas.
Esta familia de archivos está formada por versiones consultables mediante terminal de publicaciones de institutos estadísticos nacionales, o de centros de investigación universitarios, que memorizan datos recogidos bien directamente, bien utilizando como fuente los datos reunidos por diversos organismos internacionales como FAO y OCDE, dejándolos disponibles para futuras elaboraciones.
Estos archivos, destinados a usuarios especializados, a menudo asocian a los procedimientos de consulta normales (análogos a los adoptados en las bases de datos bibliográficos) ciertas funciones, a veces muy sofisticadas, para la elaboración de los datos extraídos. Así, se pueden combinar entre sí dos o más series numéricas, insertar, elaborar y conservar en una memoria especial de la serie de uso reservado, hacer la representación gráfica, interpolar datos que faltan y calcular extrapolaciones utilizando modelos matemáticos suministrados por el productor o estudiados por el usuario.
Los aspectos que en un principio han condicionado la difusión de esta clase de archivos han sido el costo de consulta, que es muy elevado si se utiliza también la potencia de cálculo disponible, y el ritmo de actualización, que en el caso de archivos con datos suministrados por instituciones externas resulta poco satisfactorio.
Actualmente, estos problemas se están reduciendo, puesto que los productores de bancos de datos han mejorado los procedimientos de adquisición de los mismos con una actualización más eficaz. Los costos siguen siendo elevados, pero pueden reducirse si la consulta la efectúa personal especializado, capaz de optimizar los recursos de cálculo disponibles y de reducir los tiempos de elaboración.

(A Beverini. Extracto de «Bancos de datos», en la publicación CELLULOSA E CARTA, n.º 4, julio-agosto de 1983).

Los pasos expuestos evidencian las partes fundamentales de un programa de Sort: comparación de datos y consiguientes desplazamientos en el seno del file de salida. El tema será ampliado más adelante, tras haber descrito las principales instrucciones Basic.
Utilizando una copia del file de datos como salida del Sort, nos encontramos con dos inconvenientes: aumenta la ocupación de la memoria masiva (al final del Sort el file de datos queda duplicado) y disminuye notablemente la velocidad de ejecución, puesto que por cada record a situar hay que desplazar los datos contenidos en el file de salida. Utilizando el método del file índice no se ocupa la memoria masiva y, sobre todo, los desplazamientos pueden efectuarse en la memoria central (el file índice no contiene datos y, por tanto, es mucho más pequeño, hasta el punto de poder alojarse en la memoria). Se consigue así un notable aumento de la velocidad de ejecución del programa de Sort.

Compactación de datos

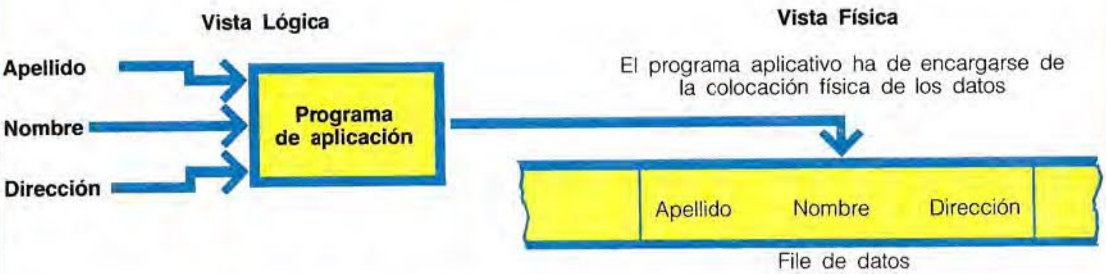
En algunas aplicaciones conviene «compactar» los datos antes de escribirlos en los soportes físicos de memorización.
En los grandes sistemas, donde la masa de datos es importante, el ahorro de algunos caracteres puede suponer una notable reducción en el espacio ocupado; en estos casos, un mejor aprovechamiento de las memorias masivas justifica una mayor complicación de los programas (hay que prever las instrucciones de compactación y desempaquetado), mientras que en los pequeños sistemas de microordenador no conviene utilizar técnicas tan complejas. Hay, sin embargo, algunos casos especiales que se comentarán más adelante. Las principales técnicas de compactación se basan en la eliminación de campos y caracteres ausentes y en la codificación de algunas combinaciones de caracteres. Las principales eliminaciones son:
Campos ausentes. Si uno de los campos de datos no se halla presente, en lugar de escribir sus espacios en blanco, conviene eliminarlo. El resultado es un record más corto que los demás. La ventaja de una menor ocupación se paga con tener records de longitud variable y, por tanto, programas más complejos.
Eliminación de caracteres inútiles. En los campos en formato ASCII se pueden eliminar los espacios en blanco entre un carácter y el siguiente; análogamente, en los campos numéricos se pueden eliminar los ceros. También en este caso hacen falta programas complejos capaces de reconocer cuántos son y dónde están los espacios o ceros eliminados para poder reconstruir íntegramente los datos en el momento de la presentación.
Codificación de combinaciones de caracteres. Algunas combinaciones de caracteres pueden ser recurrentes y, en tal caso, se sustituyen por un símbolo. Por ejemplo, sea un archivo de nómina con los apellidos Domínguez, Prado y Rodríguez.
Codificando, por ejemplo: guez = 1, ado = 2, los datos se convierten en:
Domín1, Pr2, Rodrí1.
Esta técnica resulta ventajosa sólo si las combinaciones de caracteres codificadas (1 = guez, 2 = ado) se repiten con frecuencia.

Bancos de datos en los grandes sistemas

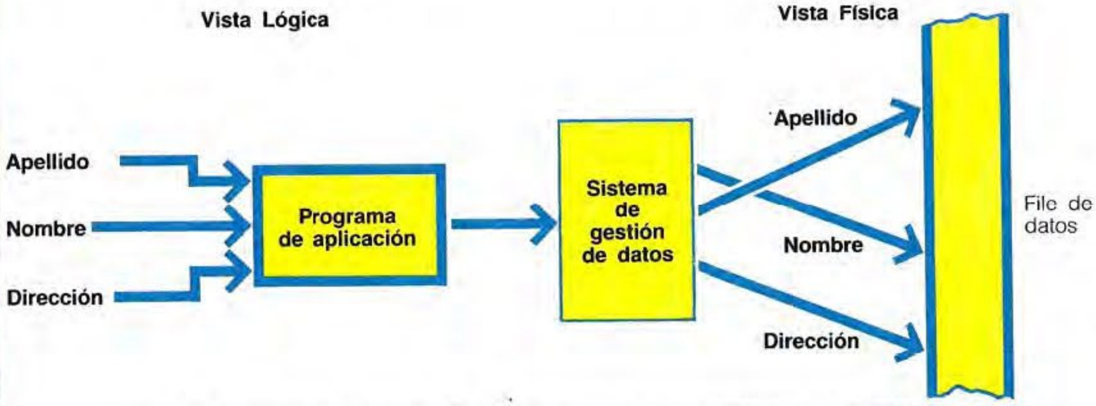
Los problemas que hemos visto para la escritura o modificación de los programas de gestión de archivos desaparecen en las aplicaciones en máquinas que son de categorías superiores.
En los ordenadores gigantes y en algunos miniordenadores hay programas muy complejos para la gestión de datos. Con estos programas el usuario ya no tiene que preocuparse de la estructura «física» de los archivos; sólo tiene que definir la «vista lógica», es decir, la estructura de los datos desde el punto de vista lógico. Será luego tarea de los programas del sistema traducir la estructura en esquemas físicos y ocuparse de su gestión. En estas aplicaciones, el programador no conoce la posición física de los datos, puesto que el espacio disponible en las memorias masivas (discos, tambores) es gestionado por el sistema. En el gráfico de la pág. 284 se muestran dos esquemas que ilustran cualitativamente la diferencia entre los dos métodos.
En el primer caso (ausencia de programas de gestión de datos), el programador ha de definir la «vista lógica», es decir, el contenido de su archivo (en el ejemplo, los datos apellido, nombre y dirección), la estructura física del record y el posicionamiento del record en el file.
En el segundo caso, con la utilización de programas de gestión de datos, el usuario sólo tiene que definir la vista lógica; la elección de cómo y dónde escribir los datos corre a cargo del sistema. En los sistemas más evolucionados, la gestión de datos puede efectuarse mediante un

lenguaje especialmente concebido, distinto de los utilizados para la programación normal, o bien con un lenguaje «huésped».
En este último caso, el lenguaje específico de gestión de datos va «unido» a un lenguaje general de programación (Basic, Cobol, etc.) y en la aplicación se utilizan las instrucciones del lenguaje específico o las del lenguaje general.
Dada la importancia de la cuestión, en 1959 se creó, en Washington, el **CODASYL** (Conference on Data System Language) para unificar los estudios realizados en todo el mundo sobre este tema y emitir normas para el desarrollo de los sistemas de **base de datos** (con este término se designa el conjunto de las vistas lógicas y de las relaciones entre los distintos campos).

GESTION DE DATOS Y DE PROGRAMAS DE APLICACION

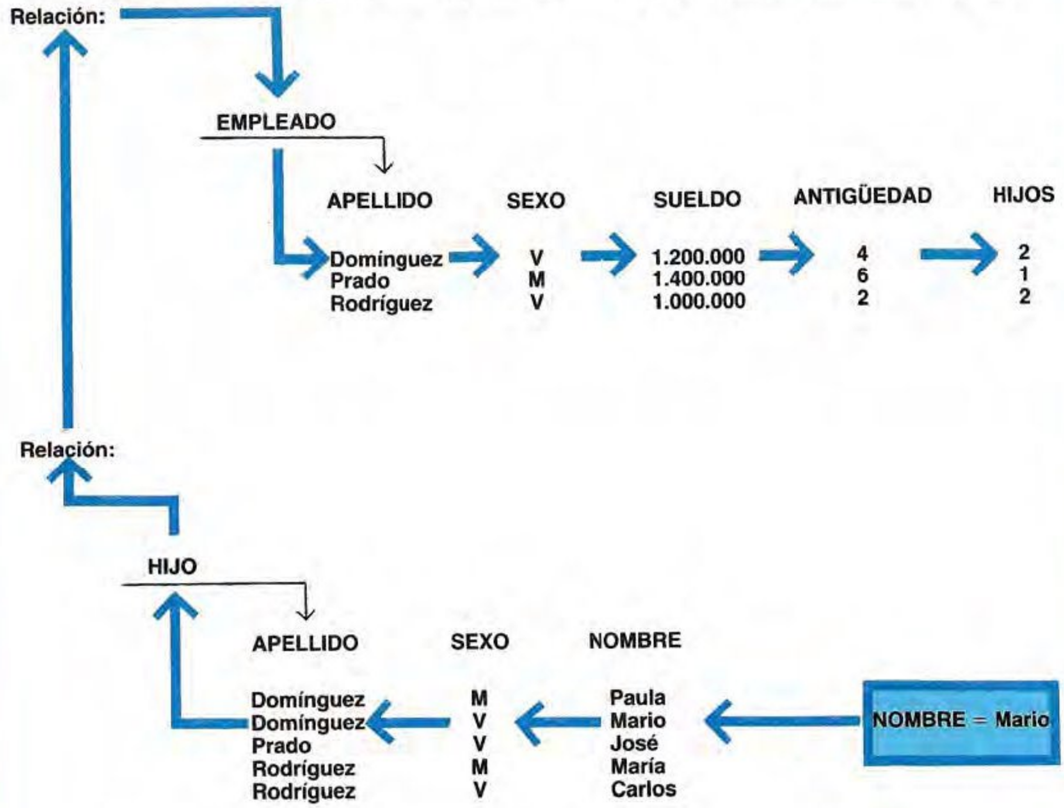


Sin el auxilio de sistemas evolucionados para la gestión de datos, todas las funciones, incluido el posicionamiento físico de los datos en las memorias masivas, han de ser realizadas por programas de aplicación



En los grandes sistemas, el uso de programas especializados en la gestión de bancos de datos modifica sustancialmente el método de programación. El usuario sólo tiene que preocuparse de definir la vista lógica de sus datos; la elección de cómo y dónde serán situados dichos datos queda completamente a cargo de los programas del sistema

EJEMPLO DE BUSQUEDA DE UNA BASE DE DATOS RELACIONAL



El dominio NOMBRE (NOMBRE = Mario) mediante las relaciones HIJO y EMPLEADO permite seleccionar todos los datos del empleado Domínguez

Base de datos relacional

Con ella se tiene una total independencia entre la estructura lógica de los datos y la física. En la base de datos relacional se define una única estructura lógica, y todos los datos están ligados mediante «relaciones».

Partiendo de cualquier dato o «dominio» (que constituye la clave de la «relación») es posible, con las relaciones que lo ligan a los demás, obtener todos los datos correlativos. En el gráfico superior se muestra un ejemplo. Los datos son los elementos de un archivo de empleados, con las dos relaciones: Empleado, Hijo; sobre un determinado valor (Mario) del dominio NOMBRE, se obtienen todos los datos del empleado.

La base de datos relacional es la forma más avanzada de gestión de datos; se suministra sólo con algunas grandes máquinas (por ej., IBM S-38) y para aplicaciones de cierta complejidad.

SOLUCIONES DEL TEST 7

1 / c: en un file secuencial los records no pueden ser leídos o escritos al azar. Los datos se memorizan por orden (secuencial) de introducción.

2 / b y c: el índice es una zona del file (o un file separado) donde se memorizan las palabras clave y el «puntero» al record que contiene los datos. Puede haber varios índices, cada uno para una posible clave de acceso al record.
Por ejemplo, en un archivo de personal que contenga los datos:
– apellido y nombre – cualificación, nivel, sección a la que pertenece – sueldo base, tasa horaria para horas extraordinarias – vacaciones, faltas, enfermedad –
puede ser útil mantener varios índices: el primero para los apellidos (que constituye la clave principal de acceso al record), y los otros para los datos secundarios, que permiten acceder al record según lógicas distintas de la búsqueda nominal.
Por ejemplo, el índice por días de vacaciones permite visualizar a todo el personal que gozará de determinado período de vacaciones; el índice por secciones puede utilizarse para tener una lista del personal de una determinada sección.
En los sistemas más evolucionados es posible la «búsqueda cruzada», es decir, se pueden combinar los campos clave con operadores lógicos adecuados. Por ejemplo, se puede seleccionar todo el personal de la «sección barnizado» que tenga 20 días de vacaciones y que no sea de nivel A. Desde el punto de vista lógico la instrucción podría responder a la fórmula: (sección = barnizado) AND (vacaciones = 20) AND (nivel ≠ A). (El símbolo ≠ significa «diferente»; en Basic, la condición «diferente» se indica con el signo < >, por lo que habría que escribir: nivel < > A.)

3 / a: es el nombre que se da a la operación de ordenación. Recordemos que la ordenación de un file consiste en extraer los records según el orden de su campo clave. En el ejemplo anterior (punto 2), con el uso de varios índices se pueden obtener listados del personal con distintas ordenaciones (una por cada campo clave). También para el Sort es posible combinar varias lógicas en una sola instrucción. Así, se puede pedir la impresión por orden de sección y, dentro de esta ordenación, por orden alfabético: en una misma sección, Prado se imprimirá antes que Rodríguez.

4 / La búsqueda en serie consiste en la lectura de todos los records de un file hasta encontrar el requerido. En la búsqueda dicotómica, la lectura del file se efectúa con sucesivas divisiones por la mitad, por lo que requiere mucho menos tiempo.
Este tipo de búsqueda se utiliza sólo en files ordenados, y antes de aplicar la búsqueda dicotómica (o binaria) hay que efectuar un Sort. Para un file con pocos datos, las operaciones Sort más búsqueda requieren el mismo tiempo que la búsqueda en serie.

5 / Esta prueba es sólo indicativa; las soluciones dependen de la longitud de los campos. Unos valores medios podrían ser:

nombre y apellido	50 caracteres	
cualificación	2 caracteres	(siglas de dos letras)
nivel	2 caracteres	(siglas)
antigüedad	2 caracteres	(de 0 a 99 años)
sueldo base	7 caracteres	(hasta 9.999.999)
días de ausencia	2 caracteres	(hasta 99)
TOTAL	65	

El record puede tener una longitud de 65 caracteres. Ya que conviene dejar un margen de espacio, por ej. 15 caracteres, la longitud del récord será de 80 caracteres. Para hacernos una idea de la capacidad de memorización de los diskettes, calculemos cuántos registros nominativos pueden memorizarse en uno de 250 kbytes y en otro de 1 Mbyte (kbyte = kilo-byte; Mbyte = mega-byte).
En el disco de 250k:
capacidad total = 250k = 250.000 bytes (caracteres)

$$\text{cantidad de datos (de 80 caracteres)} = \frac{250.000}{80} = 3.125$$

En el segundo caso (1 Mbyte = 1.000.000 de caracteres)

$$\text{cantidad de datos} = \frac{1.000.000}{80} = 12.500$$

Por tanto, con un solo diskette se puede conservar un archivo de personal con más de 12.000 registros nominativos.
En el ejemplo, algunos campos numéricos (sueldo base) se han considerado caracteres, ya que los sistemas operativos más conocidos para la transferencia a disco utilizan (en Basic) el formato en caracteres también para los números. Algunas máquinas pueden utilizar el formato numérico (un número es tratado como tal y no como una serie de caracteres). En este caso se ahorra espacio, puesto que el formato numérico (binario) ocupa menos que el formato por caracteres (ASCII). Por ejemplo, el número 60 en binario ocupa sólo 8 bits (1 byte), puesto que 60 hexadecimal = 0111100, mientras que expresando el número 60 en ASCII hacen falta dos bytes, uno para el 6 y otro para el 0. El número 60 considerado como caracteres, o sea representado en ASCII, se convierte en:
6 = 0110110 y 0 = 0110000 (ver en pág. 110 el capítulo de los códigos ASCII).

6 / El ejemplo es análogo al anterior. Una forma de estructura puede ser:

artículo	20 caracteres (descripción del artículo)
existencias	4 caracteres (hasta 9.999)
reserva	4 caracteres
fecha	8 caracteres (día = 2 caracteres, mes = 2, año = 4, por ej. 21111983)
proveedor	20 caracteres
TOTAL	56 caracteres, más un margen, 70 caracteres

7 / En el file anterior se puede ahorrar espacio de memorización de dos formas muy simples. La primera consiste en codificar la fecha indicando sólo las dos últimas cifras: 83 equivale a 1983. La segunda consiste en indicar con 1 al proveedor López, con 2 a Pérez, etc. Con esta codificación, en lugar del nombre del proveedor se introduce un código, o sea un valor numérico comprendido entre 1 y 100. El número máximo de proveedores es 100; pero el número 100 tiene tres cifras, y con tres cifras se puede escribir hasta 999, por lo que los proveedores posibles son 999. Con estos códigos se logra un ahorro de 2 bytes en la fecha y de 17 en el proveedor; la longitud del récord se reduce a 37 (con margen, 50). Un file de este tipo puede contener alrededor de 5.000 denominaciones en los diskettes más pequeños y 20.000 en los mayores.

GLOSARIO

Principales términos utilizados en la gestión de archivos

Acceso. Operación de lectura o escritura de datos en las memorias masivas (discos fijos, floppy, etc.).

Algoritmo. Método de cálculo utilizado para la resolución de un determinado problema.

Atributos. Características de los campos de datos. Por ejemplo, un campo que contenga cifras tiene el atributo de ser numérico.

Banco de datos. Conjunto de datos relativos a algunos temas concretos. En su acepción más común, los bancos de datos se identifican por dos características fundamentales. Ante todo, la información obtenida con la consulta coincide con el dato final buscado. No incluye, pues, ningún tipo de publicación secundaria, sino que están formados por copias de documentos originales consultables mediante terminal. La otra característica es la naturaleza numérica del contenido, formado por series históricas y tablas.

Base de datos. Conjunto de datos organizados según un mismo esquema lógico. El término tiene un significado distinto que «banco de datos» (que puede responder a más de un esquema lógico), aunque a veces, impropiamente, ambos términos se confunden. Por «base de datos» se entiende a menudo los archivos donde se memorizan, de forma consultable mediante terminal, las denominadas «publicaciones secundarias». Con este término se designan las recensiones, en forma unificada y sintética, de revistas, monografías, libros, informes de conferencias, etc., destinadas a suministrar a una demanda muy diversificada un medio eficaz para la localización de documentos relacionados con una consulta específica.

Buffer. Area de memoria utilizada como soporte momentáneo de los datos. En los microordenadores, durante las operaciones con empleo de memoria masiva los datos transitan siempre por un buffer. Por ejemplo, en la fase de escritura en floppy, los datos pasan de la memoria central al buffer y de éste al floppy.

Búsqueda binaria. Método de búsqueda de datos basado en sucesivas divisiones por 2.

Campo. Dato elemental. Por ejemplo, en un record de almacén, uno de los campos es la descripción del artículo, otro el costo, etc.

Clave. El campo de los datos utilizado para su búsqueda.

Compactación. Técnica mediante la cual se puede reducir el número de bits necesarios para escribir un dato.

DAM (Direct Access Method). Método de acceso directo.

Diccionario. Conjunto de nombres previstos en un archivo.

Direccionamiento indirecto. Consiste en hallar un dato mediante un file de índices que contiene los punteros. Es una búsqueda a dos niveles; el primer puntero selecciona un record del índice, que contiene el puntero a los datos.

Directorio. Indice de los datos o de los files existentes y sus características físicas (longitud y posición en las memorias masivas).

Encadenar (Chain). Cada record contiene un puntero qu señala otro record (concatenado con el primero). Esta técnica, utilizada normalmente en los bancos de datos, permite construir una «cadena» de punteros que partien de un dato localizan todos los que están conectados con aunque estén escritos en distintas posiciones de la memoria masiva.

File. Espacio físico y lógico destinado a acoger un conjur de datos homogéneos.

Indice. Valor numérico que suministra la posición de un record.

ISAM (Index Sequential Access Method). Método de acceso secuencial con índices.

Memoria virtual. Una memoria que, en el programa pare más amplia de lo que en realidad es porque hay un intercambio tan rápido de bloques de datos entre la memoria masiva y el programa, que parecen contenidos éste.

Merge. Este término indica la operación (y por extensión programa de gestión que la efectúa) de fusión de los dat contenidos en dos archivos distintos.

Método de acceso. Técnica mediante la cual se leen o escriben los datos. Los principales métodos de acceso s el aleatorio (random), el secuencial y el secuencial con índices (ISAM).

Puntero. Valor numérico que indica la posición de un record o de un dato.

Random (azar). Ver «Método de acceso».

Relacional (Base de datos relacional). Estructura lógica datos en la que los campos (dominios) están ligados por medio de «relaciones» que los conectan lógicamente.

SAM (Secuential Access Method). Método de acceso secuencial.

Sector. División física del disco. El sector es la zona míni que puede ser direccionada por el sistema operativo en l operaciones de lectura/escritura.

Sort. Ordenación de un file según una clave determinada

Tiempo de acceso. Tiempo necesario para conseguir un dato. Se compone del tiempo necesario para el posicionamiento mecánico de la cabeza de lectura y del tiempo de transferencia del dato de la memoria masiva a de la unidad central.

VTOC (Volume Tabla Of Content). Lista del contenido de disco.

de

ue

do
él,

nto

ece

en

el
os

on

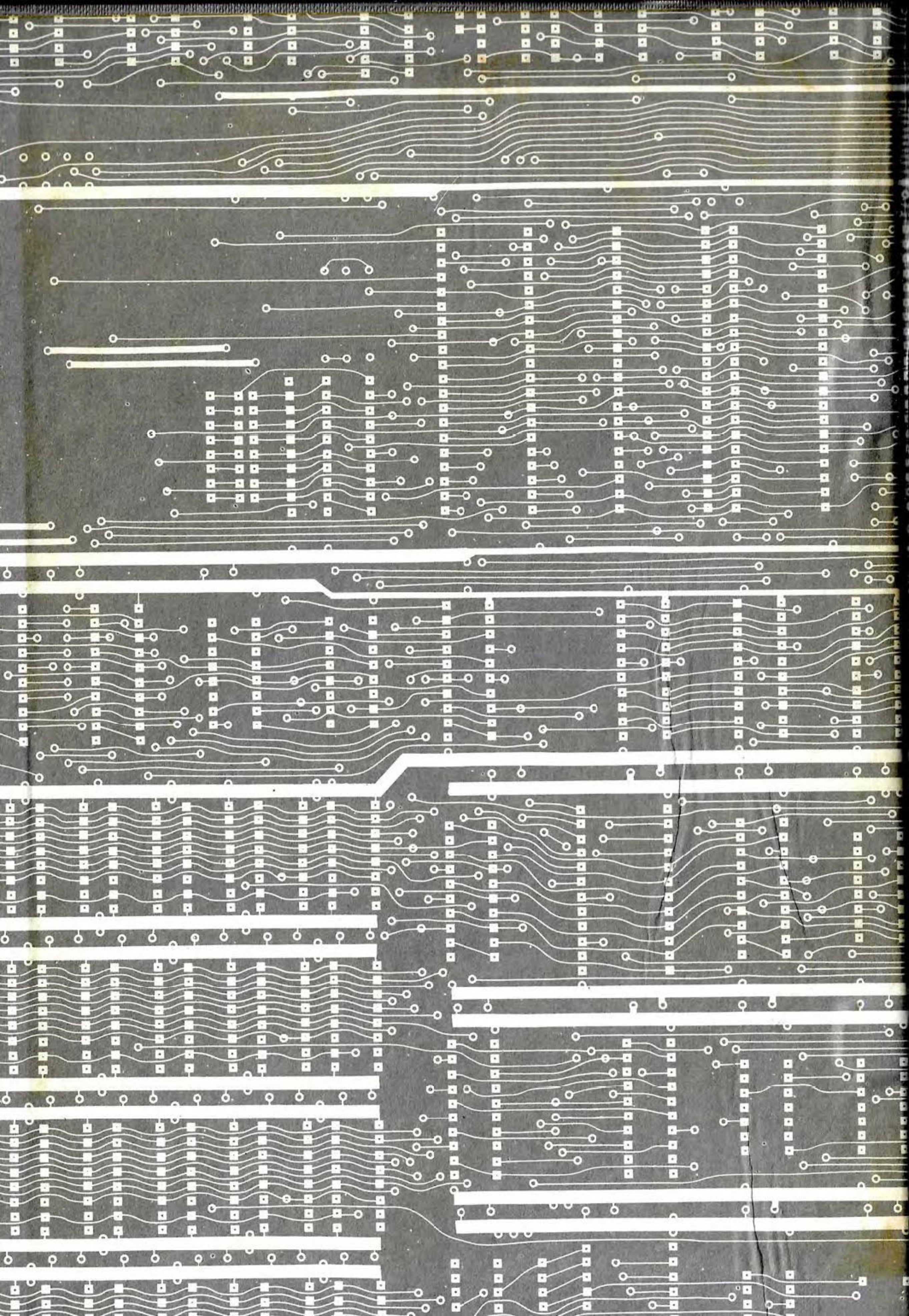
de

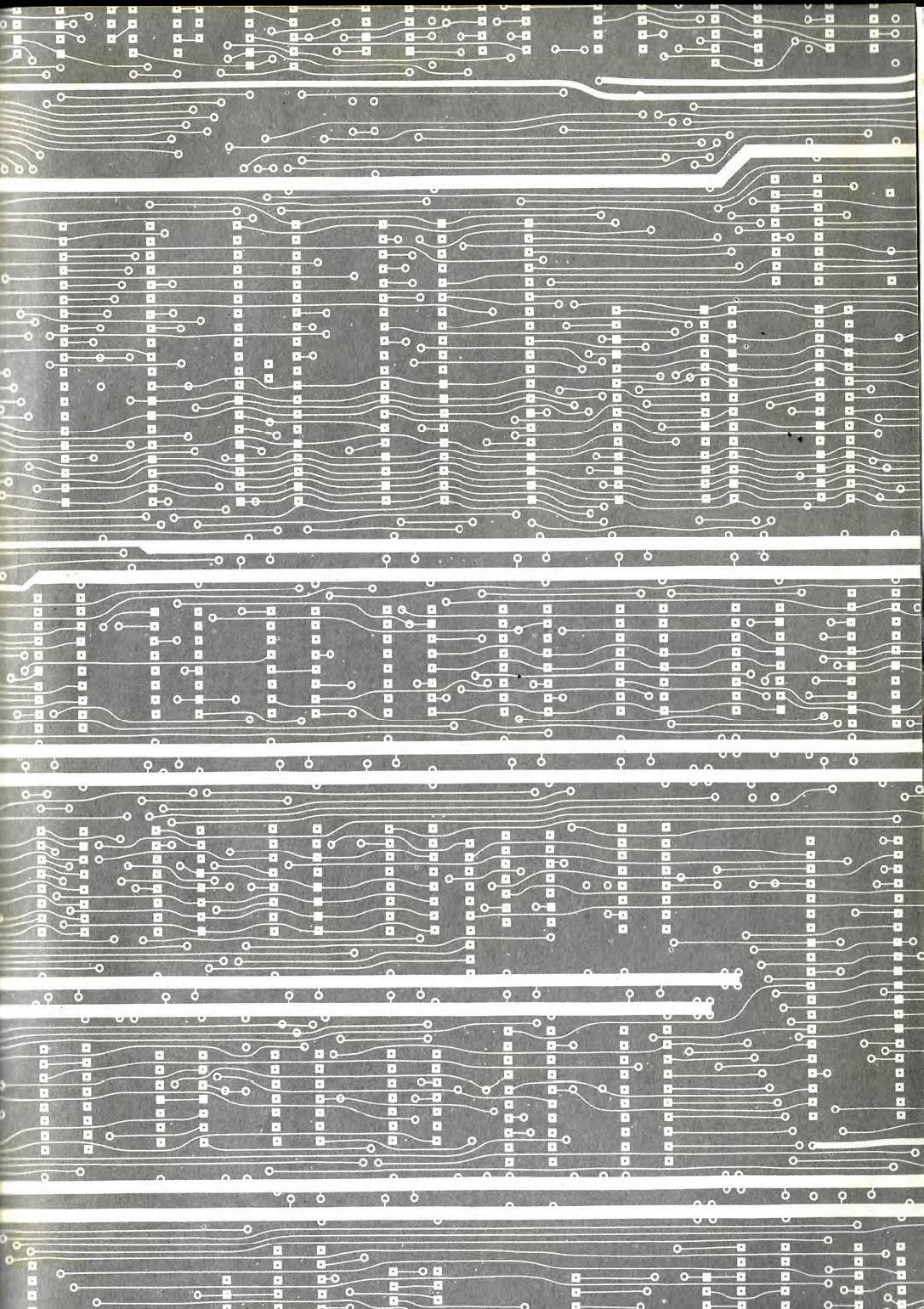
ma
as

.

la

un





EDICIONES
FORUM

1

BASIC

ENCICLOPEDIA DE LA INFORMATICA
MINIORDENADORES Y ORDENADORES PERSONALES

